On Using Certified Training towards Empirical Robustness

Anonymous authors
Paper under double-blind review

Abstract

Adversarial training is arguably the most popular way to provide empirical robustness against specific adversarial examples. While variants based on multi-step attacks incur significant computational overhead, single-step variants are vulnerable to a failure mode known as catastrophic overfitting, which hinders their practical utility for large perturbations. A parallel line of work, certified training, has focused on producing networks amenable to formal guarantees of robustness against any possible attack. However, the wide gap between the best-performing empirical and certified defenses has severely limited the applicability of the latter. Inspired by recent developments in certified training, which rely on a combination of adversarial attacks with network over-approximations, and by the connections between local linearity and catastrophic overfitting, we present experimental evidence on the practical utility and limitations of using certified training towards empirical robustness. We show that, when tuned for the purpose, a recent certified training algorithm can prevent catastrophic overfitting on single-step attacks, and that it can bridge the gap to multi-step baselines under appropriate experimental settings. Finally, we present a novel regularizer for network over-approximations that can achieve similar effects while markedly reducing runtime.

1 Introduction

The discovery of adversarial examples (Biggio et al., 2013; Szegedy et al., 2014; Goodfellow et al., 2015), semantic invariant perturbations that induce high-confidence misclassifications in neural networks, has led to the development of a variety of adversarial attacks (Moosavi-Dezfooli et al., 2016; Carlini & Wagner, 2017) and empirical defenses (Papernot et al., 2016; Cisse et al., 2017; Tramèr et al., 2018). Adversarial training (Madry et al., 2018) is undeniably the most successful empirical defense, owing to its efficacy and conceptual simplicity. Employing a robust optimization perspective, Madry et al. (2018) train using the loss incurred at the point returned by a multi-step attack, named PGD. The large cost of PGD-based adversarial training ushered in the development of less expensive techniques (Shafahi et al., 2019). However, single-step attacks (Goodfellow et al., 2015), were shown to suffer from a phenomenon known as Catastrophic Overfitting (CO) (Wong et al., 2020), under which they display a vulnerability to multi-step attacks whilst preserving strong robustness to single-step attacks. It was shown that CO can be mitigated at no additional cost through strong noise (de Jorge et al., 2022) for smaller perturbations, yet stronger attack models require the addition of explicit regularizers based on local linearity (Andriushchenko & Flammarion, 2020; Rocamora et al., 2024). Indeed, strong links between lack of local linearity and CO exist (Ortiz-Jimenez et al., 2023).

While empirical robustness to strong attack suites Croce & Hein (2020) is understood to be sufficient for many practical use-cases, a network could still be vulnerable to stronger and unseen attacks, casting doubts over its suitability for safety-critical contexts. In light of this, a variety of techniques to formally verify network robustness have been designed (Katz et al., 2017; Bunel et al., 2018; Ehlers, 2017; Tjeng et al., 2019; Lomuscio & Maganti, 2017). In spite of significant improvements over the last years (Bunel et al., 2020b; Wang et al., 2021; Ferrari et al., 2022), neural network verification techniques still face significant scaling challenges. As a result, a series of works have devised so-called certified training schemes, which produce networks more amenable to formal verification by enforcing small network over-approximations (Gowal et al., 2018; Mirman et al., 2018; Zhang et al., 2020; Wong & Kolter, 2018; Xu et al., 2020). Remarkably, a particularly inexpensive over-approximation based on Interval Bound Propagation (IBP) has emerged as a crucial technique in the area (Jovanović et al., 2022; Mao et al., 2024b). Indeed, the current state-of-the-art in certified training relies on a combination of adversarial attacks and IBP, greatly improving performance over a variety of settings (Müller et al., 2023; Mao et al., 2023). Some of these techniques were formalized into the notion of expressive losses, capable of interpolating between attacks and over-approximations (De Palma et al., 2024b).

In spite of their commonalities, certified and adversarial training have so fare evolved as two orthogonal lines of work with distinct objectives: certified and empirical robustness, respectively. This belief is consolidated by a wide and persisting performance gap between the best-reported certified and empirical robustness for a given setup (Croce & Hein, 2020; Jovanović et al., 2022; De Palma et al., 2024b). Intrigued by the versatility of recent certified training algorithms and by the links between local linearity and small network over-approximations, we present a comprehensive empirical study on the applicability of certified training techniques towards empirical robustness, leading to the following contributions:

- We systematically show for the first time that, *when explicitly tuned for the purpose*, a multiplicative expressive loss named Exp-IBP (De Palma et al., 2024b) can effectively prevent CO on setups common in the single-step adversarial training literature (§5.1). Under a limited subset of these settings, it can match or overcome the performance of some multi-step adversarial training baselines (PGD-5) while exclusively relying on single-step attacks and enforcing large verifiability.
- We show that the relative empirical robustness of Exp-IBP improves on shallower networks and longer training schedules, demonstrating that pure IBP outperforms a strong multi-step adversarial training baseline (PGD-10) for large perturbation radii on CIFAR-10 (§5.2).
- We present a novel proxy for the IBP loss, named ForwAbs, which can be computed at the cost of a single network evaluation (§3.3). When used as a regularizer, ForwAbs can prevent CO and improve the robustness of single-step attacks while significantly reducing overhead compared to expressive losses.

We believe our work demonstrates both the potential and the limitations of using current certified training techniques towards empirical robustness, paving the way for further developments in this direction. Code is provided as part of the supplementary material.

2 Background

We will use the following notations: a for a scalar, $\mathcal{A}$ for sets, A for matrices, $\boldsymbol{x}$ for a vector and $\boldsymbol{x}_i$ for its i-th component. We will furthermore use $[\![a, b]\!]$ for integer ranges, while denoting by $\min_{\boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x})$ the vector given by the entry-wise minimization over the vector function $\boldsymbol{f}$, by $\operatorname{Proj}(\boldsymbol{x}, \mathcal{A})$ the Euclidean projection of $\boldsymbol{x}$ onto $\mathcal{A}$, and by $|A|$ the entry-wise absolute value of A. Finally, $\mathbf{1}$ stands for a vector whose entries are all 1.

2.1 Adversarial Robustness

Let $\boldsymbol{f_\theta} : \mathbb{R}^d \to \mathbb{R}^k$ be a neural network parameterized by $\boldsymbol{\theta}$. Given an input $\boldsymbol{x} \in \mathbb{R}^d$ and a label $y \in [\![0, k-1]\!]$ from a k-way classification dataset $(\boldsymbol{x}, y) \sim \mathcal{D}$, $\boldsymbol{f_\theta}$ is said to be locally robust on $\boldsymbol{x}$ if, given a set of allowed perturbations $\mathcal{C}(\boldsymbol{x}) \subseteq \mathbb{R}^d$:

$$\boldsymbol{x}' \in \mathcal{C}(\boldsymbol{x}) \implies \arg\max_i \boldsymbol{f}_\theta(\boldsymbol{x}')_i = y. \tag{1}$$

In other words, the network is robust if the predicted label for $\boldsymbol{x}$ is invariant to perturbations in $\mathcal{C}(\boldsymbol{x})$. In the following, we will focus on ℓ_∞ balls of radius ϵ: $\mathcal{C}(\boldsymbol{x}) = \mathcal{B}_\epsilon(\boldsymbol{x}) := \{\boldsymbol{x}' : ||\boldsymbol{x}' - \boldsymbol{x}||_\infty \leq \epsilon\}$.

Given a surrogate loss $\mathcal{L}$ for classification, typically cross-entropy, the problem of training a network satisfying equation (1) for all $(\boldsymbol{x}, y) \sim \mathcal{D}$, $\boldsymbol{f_\theta}$ can be formulated as a robust optimization problem (Madry et al., 2018):

$$\min_{\boldsymbol{\theta}} \left[\underset{(\boldsymbol{x},y)\in\mathcal{D}}{\mathbb{E}} \mathcal{L}^*(\boldsymbol{f}_\theta, \mathcal{B}_\epsilon(\boldsymbol{x}); y) \right], \quad \text{where} \;\; \mathcal{L}^*(\boldsymbol{f}_\theta, \mathcal{B}_\epsilon(\boldsymbol{x}); y) := \max_{\boldsymbol{x}' \in \mathcal{B}_\epsilon(\boldsymbol{x})} \mathcal{L}(\boldsymbol{f}_\theta(\boldsymbol{x}'), y).$$

Computing the worst-case loss $\mathcal{L}^*(\boldsymbol{f}_\theta, \mathcal{B}_\epsilon(\boldsymbol{x}); y)$ exactly is not feasible at training time as, owing to the non-linearity of the network activations, it involves computing the global optimum of a non-concave optimization problem. In the case of piecewise-linear networks, this was shown to be NP-complete (Katz et al., 2017). The worst-case loss is hence commonly replaced by different approximations depending on the training goal.

2.2 Training for Empirical Robustness: Adversarial Training

Adversarial attacks are heuristic algorithms, typically based on gradient-based local maximization (Madry et al., 2018) or random search (Andriushchenko et al., 2020), that provide lower bounds to $\mathcal{L}^*(\boldsymbol{f}_\theta, \mathcal{B}_\epsilon(\boldsymbol{x}); y)$. Let us denote by $\boldsymbol{x}_{\text{adv},\mathbb{A}}$ the output of running an attack $\mathbb{A}$ on $\boldsymbol{f}_\theta$ for the perturbation set $\mathcal{B}_\epsilon(\boldsymbol{x})$. A network is deemed to be *empirically robust* to $\mathbb{A}$ for $(\boldsymbol{x}, y)$ if $\arg\max_i \boldsymbol{f}_\theta(\boldsymbol{x}_{\text{adv},\mathbb{A}})_i = y$. When training for empirical robustness, $\mathcal{L}^*(\boldsymbol{f}_\theta, \mathcal{B}_\epsilon(\boldsymbol{x}); y)$ is replaced by the so-called adversarial loss:

$$\mathcal{L}_{\text{adv}}(\boldsymbol{f}_\theta, \mathcal{B}_\epsilon(\boldsymbol{x}); y) := \mathcal{L}(\boldsymbol{f}_\theta(\boldsymbol{x}_{\text{adv},\mathbb{A}}), y),$$

which is a lower bound to $\mathcal{L}^*(\boldsymbol{f}_\theta, \mathcal{B}_\epsilon(\boldsymbol{x}); y)$. One of the most popular and effective adversarial training algorithms employs the PGD attack (Madry et al., 2018) at training time. Starting from a random uniform input in $\mathcal{B}_\epsilon(\boldsymbol{x})$, PGD iteratively steps using the sign of the network input gradient $\operatorname{sign}(\nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{f}_\theta(\boldsymbol{x}), \boldsymbol{y}))$, projecting back to $\mathcal{B}_\epsilon(\boldsymbol{x})$ after every iteration. Owing to the large cost of training with multi-step PGD, a series of works have focused on designing effective single-step alternatives (Shafahi et al., 2019; Wong et al., 2020; de Jorge et al., 2022). Among them is FGSM (Goodfellow et al., 2015), which systematically lands on a corner of the perturbation space: $\boldsymbol{x}_{\text{adv,FGSM}} = \boldsymbol{x} + \epsilon \ \operatorname{sign}(\nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{f}_\theta(\boldsymbol{x}), \boldsymbol{y}))$. FGSM was shown to suffer from a failure mode known as *catastrophic overfitting* (Wong et al., 2020), under which the network's robustness to PGD attacks rapidly drops to $\approx 0\%$. Wong et al. (2020) empirically demonstrate that catastrophic overfitting can be prevented at moderate ϵ values if using a random uniform input in $\mathcal{B}_\epsilon(\boldsymbol{x})$ as a FGSM starting point (RS-FGSM). de Jorge et al. (2022) propose to sample uniformly from a larger set than the target model (typically, $\mathcal{B}_{2\epsilon}(\boldsymbol{x})$), and by removing RS-FGSM's projection onto $\mathcal{B}_\epsilon(\boldsymbol{x})$ after the step. The resulting attack, named N-FGSM, attains non-negligible PGD robustness at larger ϵ without incurring any additional cost with respect to RS-FGSM, and acts as an implicit loss regularization (de Jorge et al., 2022). Nevertheless, explicit regularizers, often enforcing local linearity (Andriushchenko & Flammarion, 2020; Rocamora et al., 2024) are needed to further mitigate CO against stronger attacks.

2.3 Training for Verified Robustness: Certified Training

As described in \$2.2, the notion of empirical robustness is specific to a given attack $\mathbb{A}$. However, $\boldsymbol{f}_\theta(\boldsymbol{x}_{\text{adv},\mathbb{A}})_i = y$ does not necessarily imply that equation (1) holds, as the network may still be vulnerable to a more effective or unseen attack $\mathbb{A}'$. Formal verification provides a stronger, attack-independent, notion of robustness. A network $\boldsymbol{f}_\theta$ is said to be *certifiably robust* for $(\boldsymbol{x}, y)$ if and only if the difference between the ground truth logit $\boldsymbol{f}_\theta(\boldsymbol{x}')_y$ and the other logits is positive for all $\boldsymbol{x}' \in \mathcal{B}_\epsilon(\boldsymbol{x})$:

$$\min_i \left\{ \boldsymbol{z}_{\boldsymbol{f_\theta}}^{\mathcal{B}_\epsilon(\boldsymbol{x}),y} := \min_{\boldsymbol{x}' \in \mathcal{B}_\epsilon(\boldsymbol{x})} \left[\boldsymbol{z}_{\boldsymbol{f_\theta}}(\boldsymbol{x}, y) := (\boldsymbol{f}_\theta(\boldsymbol{x}')_y - \boldsymbol{f}_\theta(\boldsymbol{x}')) \right] \right\}_i \geq 0. \tag{2}$$

Assuming the use of a loss function $\mathcal{L}$ that is translation invariant with respect to the network output (Wong et al., 2020), such as cross-entropy, then $\mathcal{L}^*(\boldsymbol{f}_\theta, \mathcal{B}_\epsilon(\boldsymbol{x}); y) = \mathcal{L}\left(-\boldsymbol{z}_{\boldsymbol{f_\theta}}^{\mathcal{B}_\epsilon(\boldsymbol{x}),y}, y\right)$. As for $\mathcal{L}^*(\boldsymbol{f}_\theta, \mathcal{B}_\epsilon(\boldsymbol{x}); y)$ (see §2.1), computing the exact optimal logit differences $\boldsymbol{z}_{\boldsymbol{f_\theta}}^{\mathcal{B}_\epsilon(\boldsymbol{x}),y}$ is computationally infeasible. However, given lower bounds $\underline{\mathbf{z}}_{\boldsymbol{f_\theta}}^{\mathcal{B}_\epsilon(\boldsymbol{x}),y} \leq \boldsymbol{z}_{\boldsymbol{f_\theta}}^{\mathcal{B}_\epsilon(\boldsymbol{x}),y}$ to the optimal logit differences, we can conclude that $\boldsymbol{f}_\theta$ is certifiably robust if $\min_i \left(\underline{\mathbf{z}}_{\boldsymbol{f_\theta}}^{\mathcal{B}_\epsilon(\boldsymbol{x}),y}\right)_i \geq 0$. We will henceforth call a NN verifier any method to compute the above lower bounds. Given an efficient verifier, and assuming that the loss is monotonically increasing for all the network outputs except the one associated with the ground truth, networks can be trained for certified robustness by using the loss induced by the lower bounds to the logit differences (Wong & Kolter, 2018; Zhang et al., 2020), which upper bounds the worst-case loss:

$$\mathcal{L}_{\text{ver}}(\boldsymbol{f}_\theta, \mathcal{B}_\epsilon(\boldsymbol{x}); y) := \mathcal{L}\left(-\underline{\mathbf{z}}_{\boldsymbol{f_\theta}}^{\mathcal{B}_\epsilon(\boldsymbol{x}),y}, y\right) \geq \mathcal{L}^*(\boldsymbol{f}_\theta, \mathcal{B}_\epsilon(\boldsymbol{x}); y). \tag{3}$$

Networks trained via $\mathcal{L}_{\text{ver}}(\boldsymbol{f}_\theta, \mathcal{B}_\epsilon(\boldsymbol{x}); y)$ feature tight $\underline{\mathbf{z}}_{\boldsymbol{f_\theta}}^{\mathcal{B}_\epsilon(\boldsymbol{x}),y}$ lower bounds by design. We will focus on the least expensive bounding algorithm: IBP (Mirman et al., 2018; Gowal et al., 2018). When used for training, IBP outperforms tighter relaxations as it results in better-behaved loss functions (Jovanović et al., 2022).

2.3.1 IBP

IBP is a method to over-approximate neural network outputs based on the application of interval arithmetics (Sunaga, 1958; Moore, 1966) onto the functions composing the neural architecture. As such, it can be easily applied onto general computational graphs (Xu et al., 2020). For ease of presentation, let us assume that $\boldsymbol{f_\theta}$ is a n-layer feed-forward neural network, with layer $j \in [\![1, n-1]\!]$ composed of an affine operation followed by a monotonically increasing element-wise activation function $\boldsymbol{\sigma}$ such as ReLU, $\boldsymbol{x}^j = \boldsymbol{\sigma}\left(W^i \boldsymbol{x}^{j-1} + \boldsymbol{b}^i\right)$, and a final affine layer: $\boldsymbol{f_\theta} = \left(W^n \boldsymbol{x}^{n-1} + \boldsymbol{b}^n\right)$. In this case, we can compose the logit differences and the last layer into a single affine layer: $\boldsymbol{z}_{\boldsymbol{f_\theta}}(\boldsymbol{x}, y) = \left(\tilde{W}^n \boldsymbol{x}^{n-1} + \tilde{\boldsymbol{b}}^n\right)$. Then, for perturbation $\mathcal{B}_\epsilon(\boldsymbol{x}^0)$, IBP computes $\underline{\mathbf{z}}_{\boldsymbol{f_\theta}}^{\mathcal{B}_\epsilon(\boldsymbol{x}),y}$ through the following procedure, which iteratively derives lower and upper bounds $\hat{\boldsymbol{l}}^k$ and $\hat{\boldsymbol{u}}^k$ to the outputs of the k-th affine layer:

$$
\begin{gathered}
\underline{\mathbf{z}}_{\boldsymbol{f_\theta}}^{\mathcal{B}_\epsilon(\boldsymbol{x}),y} = \tfrac{1}{2}\tilde{W}^n\left(\hat{\boldsymbol{l}}^{n-1} + \hat{\boldsymbol{u}}^{n-1}\right) - \tfrac{1}{2}|\tilde{W}^n|\left(\hat{\boldsymbol{u}}^{n-1} - \hat{\boldsymbol{l}}^{n-1}\right) + \tilde{\boldsymbol{b}}^n, \quad \text{where: } \begin{bmatrix} \hat{\boldsymbol{l}}^1 = W^1\boldsymbol{x}^0 - \epsilon\left|W^1\right|\mathbf{1} + \boldsymbol{b}^1 \\ \hat{\boldsymbol{u}}^1 = W^1\boldsymbol{x}^0 + \epsilon\left|W^1\right|\mathbf{1} + \boldsymbol{b}^1 \end{bmatrix}, \\
\begin{bmatrix} \hat{\boldsymbol{l}}^j = \tfrac{1}{2}W^j\left(\boldsymbol{\sigma}\left(\hat{\boldsymbol{l}}^{j-1}\right) + \boldsymbol{\sigma}\left(\hat{\boldsymbol{u}}^{j-1}\right)\right) - \tfrac{1}{2}|W^j|\left(\boldsymbol{\sigma}\left(\hat{\boldsymbol{u}}^{j-1}\right) - \boldsymbol{\sigma}\left(\hat{\boldsymbol{l}}^{j-1}\right)\right) + \boldsymbol{b}_j \\ \hat{\boldsymbol{u}}^j = \tfrac{1}{2}W^j\left(\boldsymbol{\sigma}\left(\hat{\boldsymbol{l}}^{j-1}\right) + \boldsymbol{\sigma}\left(\hat{\boldsymbol{u}}^{j-1}\right)\right) + \tfrac{1}{2}|W^j|\left(\boldsymbol{\sigma}\left(\hat{\boldsymbol{u}}^{j-1}\right) - \boldsymbol{\sigma}\left(\hat{\boldsymbol{l}}^{j-1}\right)\right) + \boldsymbol{b}_j \end{bmatrix} \forall\ j \in [\![2, n-1]\!].
\end{gathered}
\tag{4}
$$

We will henceforth write $\boldsymbol{l}_{\boldsymbol{f_\theta}}^{\mathcal{B}_\epsilon(\boldsymbol{x}),y}$ for the lower bounds to the logit differences obtained through equation (4), and $\mathcal{L}_{\text{IBP}}(\boldsymbol{f}_\theta, \mathcal{B}_\epsilon(\boldsymbol{x}); y) := \mathcal{L}\left(-\boldsymbol{l}_{\boldsymbol{f_\theta}}^{\mathcal{B}_\epsilon(\boldsymbol{x}),y}, y\right)$ for the associated loss.

2.3.2 Hybrid Methods

A network trained via the IBP loss $\mathcal{L}_{\text{IBP}}(\boldsymbol{f}_\theta, \mathcal{B}_\epsilon(\boldsymbol{x}); y)$ can be easily verified to be robust using the same bounds employed for training. However, the relative looseness of the over-approximation from equation (4) results in a significant decrease in standard performance. For ReLU networks, which are the focus of this work, the availability of more effective verifiers based on branch-and-bound (Henriksen & Lomuscio, 2021; Wang et al., 2021; Ferrari et al., 2022) post-training has motivated the development of alternative techniques. A series of methods have shown that combination of adversarial training with network over-approximations (Balunovic & Vechev, 2020; Fan & Li, 2021; De Palma et al., 2022; Mao et al., 2023) can yield strong verifiability via branch-and-bound while reducing the impact on standard performance. In particular, Müller et al. (2023) proposed to compute the IBP loss over a tunable subset of $\mathcal{B}_\epsilon(\boldsymbol{x})$ containing adversarial examples, yielding favorable trade-offs between standard performance and certified robustness via branch-and-bound-based methods. De Palma et al. (2024b) introduced the notion of *loss expressivity*, defined as the ability of a loss function to spawn a continuous range of trade-offs between the adversarial and the IBP losses, showing that expressive losses obtained through convex combinations between an adversarial and an over-approximation component lead to state-of-the-art certified robustness.

While past work in certified training has designed and deployed these algorithms to maximize certified robustness, empirical robustness through strong attacks (Croce & Hein, 2020) remains the most adopted metric owing to the lack of scalability and to the looseness of network verifiers. Noting the potential versatility of recent hybrid certified training schemes, this work investigates whether they can be employed towards improving empirical robustness.

3 Certified Training for Empirical Robustness

Recent certified training techniques (§2.3.2) have demonstrated that the ability to precisely control the tightness of network over-approximations while preserving an adversarial training component is crucial to maximize certified robustness across experimental setups. We hypothesize that this versatility can be alternatively leveraged towards improving the empirical robustness of the adversarial training schemes on top of which they are applied, providing experimental evidence in §5. This section details the certified training algorithms employed in the experimental study. We begin with a motivating example (§3.1), then study the qualitative behavior of existing methods in settings of interest (§3.2), and conclude by presenting a novel regularizer designed to mimic the effect of IBP-based training while reducing the associated overhead (§3.3).

3.1 Motivating Example

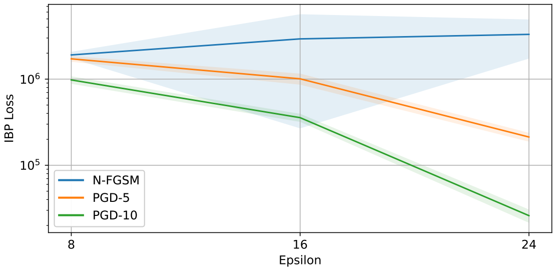


Figure 1: IBP loss of adversarial training schemes on CIFAR-10, setup from table 1.

In order to motivate our empirical study, we show that multi-step attacks, which typically exhibit superior robustness at the expense of training time, are associated to smaller network over-approximations, as measured by the IBP loss. Figure 1, which is associated to the experimental setup from table 1, shows that the IBP loss indeed decreases with the number of attack steps. Furthermore, it shows two distinct qualitative trends: for N-FGSM, which is prone to CO (causing the large experimental variability), the IBP loss increases with the perturbation radius. On the other hand, $\mathcal{L}_{\text{IBP}}(\boldsymbol{f}_\theta, \mathcal{B}_\epsilon(\boldsymbol{x}); y)$ decreases with ϵ for PGD-5 and PGD-10. Indeed, multi-step attacks yield networks that are more locally linear compared to single-step attacks (Andriushchenko & Flammarion, 2020; Rocamora et al., 2024; Ortiz-Jimenez et al., 2023), which in turn results in tighter $\underline{\mathbf{z}}^{\mathcal{B}_\epsilon(\boldsymbol{x}),y}_{\boldsymbol{f_\theta}}$ bounds and hence smaller network over-approximations. In this work, we investigate whether directly controlling $\underline{\mathbf{z}}^{\mathcal{B}_\epsilon(\boldsymbol{x}),y}_{\boldsymbol{f_\theta}}$ via certified training can benefit empirical robustness.

3.2 Expressive Losses

For a fixed verifier and attack algorithm, an expressive loss $\mathcal{L}_\alpha(\boldsymbol{f}_\theta, \mathcal{B}_\epsilon(\boldsymbol{x}); y)$ can interpolate from the adversarial loss $\mathcal{L}_{\text{adv}}(\boldsymbol{f}_\theta, \mathcal{B}_\epsilon(\boldsymbol{x}); y)$ to the worst-case loss over-approximation $\mathcal{L}_{\text{ver}}(\boldsymbol{f}_\theta, \mathcal{B}_\epsilon(\boldsymbol{x}); y)$ by continuously increasing a parameter α, the over-approximation coefficient, from 0 to 1. Formally (De Palma et al., 2024b, Definition 3.1), assuming that IBP is the verifier of choice: $\mathcal{L}_\alpha(\boldsymbol{f}_\theta, \mathcal{B}_\epsilon(\boldsymbol{x}); y)$ is monotonically increasing and continuous with respect to α, $\mathcal{L}_0(\boldsymbol{f}_\theta, \mathcal{B}_\epsilon(\boldsymbol{x}); y) = \mathcal{L}_{\text{adv}}(\boldsymbol{f}_\theta, \mathcal{B}_\epsilon(\boldsymbol{x}); y)$, $\mathcal{L}_1(\boldsymbol{f}_\theta, \mathcal{B}_\epsilon(\boldsymbol{x}); y) = \mathcal{L}_{\text{IBP}}(\boldsymbol{f}_\theta, \mathcal{B}_\epsilon(\boldsymbol{x}); y)$, and:

$$\mathcal{L}_{\text{adv}}(\boldsymbol{f}_\theta, \mathcal{B}_\epsilon(\boldsymbol{x}); y) \leq \mathcal{L}_\alpha(\boldsymbol{f}_\theta, \mathcal{B}_\epsilon(\boldsymbol{x}); y) \leq \mathcal{L}_{\text{IBP}}(\boldsymbol{f}_\theta, \mathcal{B}_\epsilon(\boldsymbol{x}); y) \;\; \forall\; \alpha \in [0, 1].$$

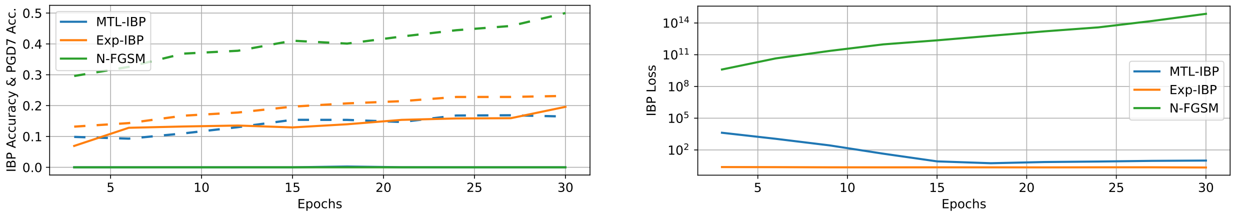
As shown by De Palma et al. (2024b) expressive losses include SABR (Müller et al., 2023) and three loss functions defined through convex combinations: CC-IBP, MTL-IBP and Exp-IBP. We found MTL-IBP, CC-IBP and SABR to display similar behaviors in the settings of interest: we refer the interested reader to appendix A for additional details. For simplicity and brevity, we will hence focus on MTL-IBP and Exp-IBP in the remainder of this work. The former consists of a convex combination of the adversarial and IBP losses. Exp-IBP, instead, performs the loss convex combination in a logarithmic space.

Definition 3.1 *The MTL-IBP and Exp-IBP losses are defined as follows:*

$$\begin{aligned}\mathcal{L}_\alpha^{\textit{MTL-IBP}}(\boldsymbol{f}_\theta, \mathcal{B}_\epsilon(\boldsymbol{x}); y) &:= (1-\alpha)\mathcal{L}_{adv}(\boldsymbol{f}_\theta, \mathcal{B}_\epsilon(\boldsymbol{x}); y) + \alpha\mathcal{L}_{IBP}(\boldsymbol{f}_\theta, \mathcal{B}_\epsilon(\boldsymbol{x}); y),\\ \mathcal{L}_\alpha^{\textit{Exp-IBP}}(\boldsymbol{f}_\theta, \mathcal{B}_\epsilon(\boldsymbol{x}); y) &:= \mathcal{L}_{adv}(\boldsymbol{f}_\theta, \mathcal{B}_\epsilon(\boldsymbol{x}); y)^{(1-\alpha)}\mathcal{L}_{IBP}(\boldsymbol{f}_\theta, \mathcal{B}_\epsilon(\boldsymbol{x}); y)^{\alpha}.\end{aligned} \tag{5}$$

Expressive losses for certified robustness In order to maximize certified accuracy (defined as the share of inputs verified to be robust) and stabilize training, previous work (Müller et al., 2023; De Palma et al., 2024b) deployed the losses in equation (5) under a specific set of conditions. First, shallow architectures, such as the popular 7-layer `CNN7` (Shi et al., 2021; Müller et al., 2023; Mao et al., 2023; De Palma et al., 2024b) (see appendix), which, aided by specialized initialization and regularization (Shi et al., 2021), contain the growth of the IBP bounds over consecutive layers. Second, relatively long training schedules featuring gradient clipping, a warm-up phase of standard training, and a ramp-up phase during which the perturbation radius used to compute both the attack and the IBP bounds is gradually increased from 0 to the target value ϵ. Finally, relatively large values of α to reduce the magnitude of the bounds $\underline{\mathbf{z}}^{\mathcal{B}_\epsilon(\boldsymbol{x}),y}_{\boldsymbol{f_\theta}}$ on the logit differences. In this context, all the expressive losses from equation (5) display a similar qualitative behavior, attaining roughly the same certified robustness (De Palma et al., 2024b).

Qualitative differences in adversarial training setups In adversarial training, overhead is often a major concern (Shafahi et al., 2019). Indeed, works on single-step adversarial training typically rely on deeper



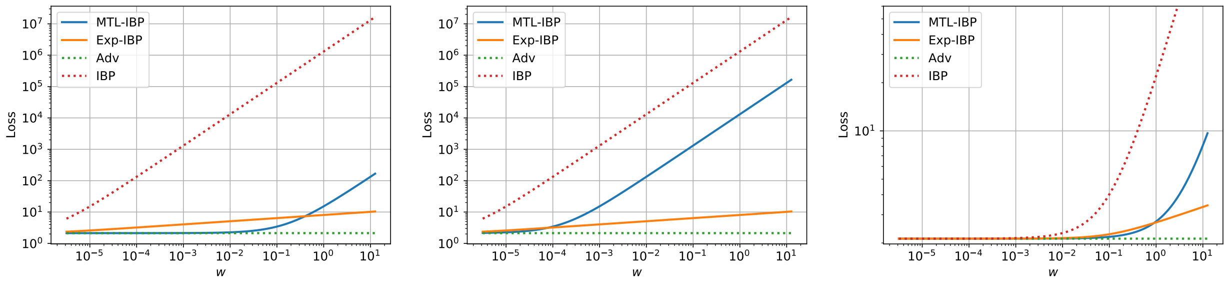
(a) Certified accuracy via IBP bounds (solid lines), empirical accuracy under PGD-7 attacks (dashed).

(b) IBP loss.

Figure 2: IBP certified robustness attained by MTL-IBP and Exp-IBP on the `PreActResNet18` training setup from de Jorge et al. (2022). Validation results on CIFAR-10 under perturbations of $\epsilon = {}^{8}/_{255}$.

networks and shorter training schedules to maximize empirical robustness while minimizing runtimes. A common setting, which we reproduce in §5.1 because of its prevalence in the catastrophic overfitting literature, involves a `PreactResNet18` (He et al., 2016) trained with a cyclic learning rate and without any gradient clipping, ramp-up (except on SVHN) nor warm-up (Andriushchenko & Flammarion, 2020; Wong et al., 2020; de Jorge et al., 2022). Owing to the exploding IBP loss at initialization and to the specific training hyper-parameters, pure IBP training is not directly applicable to this setup. In order to study the behavior of expressive losses in this context, we tuned MTL-IBP and Exp-IBP to maximize IBP-based verifiability on CIFAR-10 under perturbations of radius $\epsilon = {}^{8}/_{255}$, using N-FGSM to generate $\boldsymbol{x}_{\text{adv},\mathbb{A}}$. Figure 2 shows that MTL-IBP is unable to attain non-negligible certified accuracy via IBP in this context. On the other hand, Exp-IBP reaches almost 20% IBP certified accuracy, highlighting a significant qualitative difference in behavior. In order to gain further insight on this, we plot the sensitivity of the two expressive losses on two toy networks where the magnitude of the IBP bounds is controlled through a scalar parameter w (see appendix B.1). Figure 3 shows that on the deeper network the MTL-IBP loss either flattens onto $\mathcal{L}_{\text{adv}}(\boldsymbol{f}_\theta, \mathcal{B}_\epsilon(\boldsymbol{x}); y)$ at large IBP loss values (small α), or explodes with the IBP bounds, resulting in unstable training behavior (larger α). On the other hand, Exp-IBP can tuned so as to be able to drive $\mathcal{L}_{\text{IBP}}(\boldsymbol{f}_\theta, \mathcal{B}_\epsilon(\boldsymbol{x}); y)$ to very small values without exploding at larger w values. On the shallow network, instead, for which the IBP bounds are significantly smaller, MTL-IBP can be tuned to display roughly the same behavior as Exp-IBP for smaller $\mathcal{L}_{\text{IBP}}(\boldsymbol{f}_\theta, \mathcal{B}_\epsilon(\boldsymbol{x}); y)$ without taking overly large loss values within the considered parameter range. We believe this explains their homogeneous behavior in training setups designed to prevent large IBP bounds, such as those typically employed in the certified training literature.

Maximizing empirical accuracy As seen in figure 2, expressive losses can pay a large price in empirical accuracy compared to $\alpha = 0$ (pure N-FGSM, in this case) when α is tuned to tighten the IBP bounds. One may hence conclude that certified robustness is at odds with empirical accuracy. In section §5 we instead show that, when α is tuned for the purpose, expressive losses can result in increased empirical robustness.



(a) 18 layers; $\alpha_{\text{MTL-IBP}} = 10^{-5}$.

(b) 18 layers; $\alpha_{\text{MTL-IBP}} = 10^{-2}$.

(c) 2 layers; $\alpha_{\text{MTL-IBP}} = 3 \times 10^{-2}$.

Figure 3: Sensitivity of the expressive losses on a toy network of varying depth, with $\alpha_{\text{Exp-IBP}} = 10^{-1}$.

3.3 ForwAbs

Expressive losses require the computation of IBP bounds $\boldsymbol{l}_{\boldsymbol{f_\theta}}^{\mathcal{B}_\epsilon(\boldsymbol{x}),y}$ using the procedure in equation (4), whose cost roughly corresponds to two network evaluations (also called forward passes): one using the original network, the other employing the absolute value of the network weights. This overhead on top of adversarial training is negligible only when $\boldsymbol{x}_{\text{adv},\mathbb{A}}$ are computed via multi-step attacks. We here study less expensive ways to control the IBP bounds when single-step attacks are instead employed.

The gap between the lower and upper bounds after the k-th affine layer, which we denote by $\boldsymbol{\delta}^k := (\hat{\boldsymbol{u}}^k - \hat{\boldsymbol{l}}^k)$, can be employed as a measure of the tightness of the IBP bounds (Shi et al., 2021). For ReLU networks, $\boldsymbol{\delta}^k$ can be upper bound by $|W^k|\boldsymbol{\delta}^{k-1} \geq |W^k|(\sigma(\hat{\boldsymbol{u}}^{k-1}) - \sigma(\hat{\boldsymbol{l}}^{k-1}))$. This upper bound coincides with $\boldsymbol{\delta}^k$ in the case of Deep Linear Networks (DLNs), for which $\sigma(\boldsymbol{a}) = \boldsymbol{a}$, or if the ReLUs are locally linear ($\hat{\boldsymbol{l}}^k \geq 0$, or $\hat{\boldsymbol{u}}^k \leq 0$). We propose to employ $\bar{\boldsymbol{\delta}}^n$, which we define as the (looser) upper bound to $\boldsymbol{\delta}^n$ obtained through repeated use of the $|W^k|\boldsymbol{\delta}^{k-1}$ bound, as a regularizer for the IBP loss. The $\bar{\boldsymbol{\delta}}^n$ term can be computed at the cost of a single forward pass through an auxiliary network where each affine layer $\mathbf{g}^k(\boldsymbol{a}) = W^k\boldsymbol{a} + \boldsymbol{b}$ is replaced by $\mathbf{h}^k(\boldsymbol{a}) = |W|^k\boldsymbol{a}$. We call the resulting method ForwAbs (*Forw*ard pass in *Abs*olute value).

Definition 3.2 *The ForwAbs loss takes the following form:*

$$\begin{aligned} &\mathcal{L}_{\lambda}^{ForwAbs}(\boldsymbol{f}_\theta, \mathcal{B}_\epsilon(\boldsymbol{x}); y) := \mathcal{L}_{adv}(\boldsymbol{f}_\theta, \mathcal{B}_\epsilon(\boldsymbol{x}); y) + \lambda\left(\mathbf{1}^T\bar{\boldsymbol{\delta}}^n\right), \\ &\textit{where:} \quad \bar{\boldsymbol{\delta}}^1 = \boldsymbol{\delta}^1 = 2\epsilon|W^1|\mathbf{1}, \qquad \bar{\boldsymbol{\delta}}^k = |W^k|\bar{\boldsymbol{\delta}}^{k-1} \ \forall\, k \in [\![2, n]\!]. \end{aligned} \tag{6}$$

In order to demonstrate the empirical correlation between the ForwAbs term $\bar{\boldsymbol{\delta}}^n$ and the IBP bounds $\boldsymbol{l}_{\boldsymbol{f_\theta}}^{\mathcal{B}_\epsilon(\boldsymbol{x}),y}$, figure 4 presents the results of a ForwAbs-trained network, where the λ coefficient is tuned to minimize the IBP bounds in the same setup as figure 2. In spite of its simplicity and of its significantly smaller runtime overhead, ForwAbs can drive the IBP loss to roughly the same values attained by MTL-IBP. Furthermore, the final $\mathcal{L}_{\text{IBP}}(\boldsymbol{f}_\theta, \mathcal{B}_\epsilon(\boldsymbol{x}); y)$ attained by ForwAbs is significantly lower than those associated to N-FGSM. Strong ℓ_1 regularization over the weights (with regularization coefficient $\lambda_{\ell_1} = 0.04$) fails to achieve a comparable effect, demonstrating that the behavior of ForwAbs is not merely a consequence of small ℓ_1 norms of the network weights.

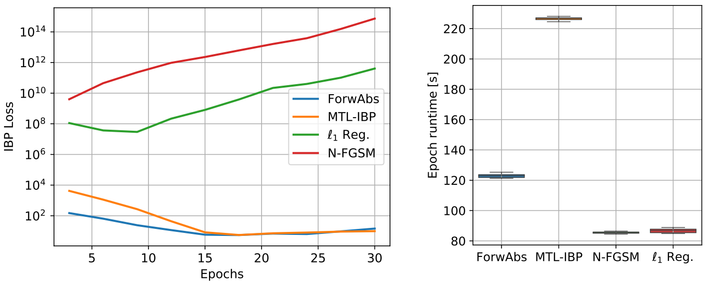


Figure 4: IBP loss over epochs (*left*), box plots (10 runs) for the training time of an epoch (*right*), setup as figure 2.

4 Related Work

We here provide an overview of closely-related work across certified and adversarial training.

4.1 Neural Network Verification and Certified Training

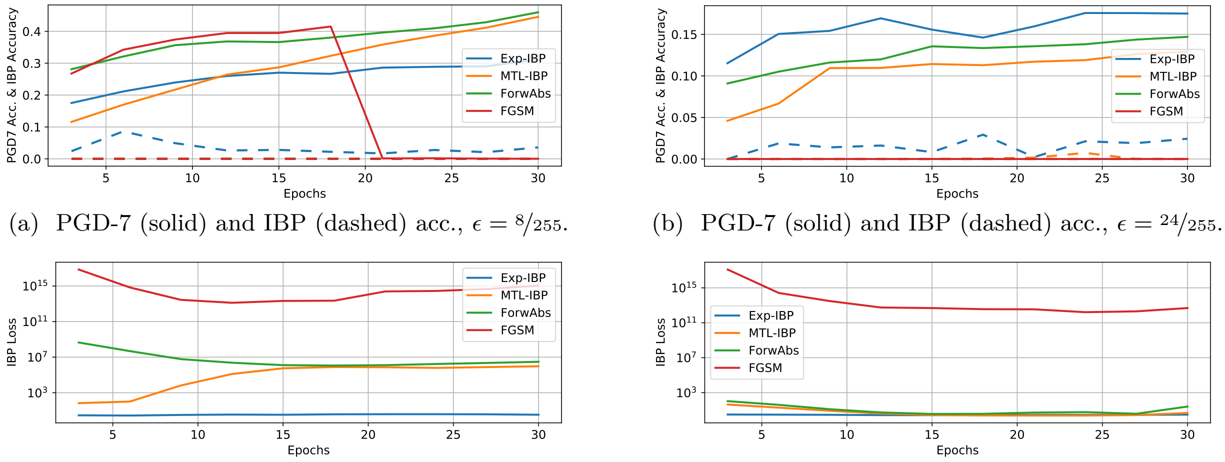
As outlined in §2.3, a neural network verifier computes lower bounds to the worst-case logit differences $\boldsymbol{z}_{\boldsymbol{f_\theta}}^{\mathcal{B}_\epsilon(\boldsymbol{x}),y}$. Given its centrality to state-of-the-art certified training schemes (§2.3.2), we have focused on the inexpensive IBP algorithm (§2.3.1). However, a significant number of verifiers exist, for which tighter lower bounds are typically associated with larger overhead. One line of work, inspired by abstract interpretation (Cousot & Cousot, 1977) builds abstract domains over-approximating the neural network (Singh et al., 2018; 2019b). Lower bounds can be also obtained by solving linear programs replacing network non-linearities with convex relaxations (Ehlers, 2017; Anderson et al., 2020; Tjandraatmadja et al., 2020; Singh et al., 2019a), potentially through dual formulations (Wong & Kolter, 2018; Dvijotham et al., 2018; Bunel et al., 2020a; De Palma et al., 2021; 2024a). Alternatively, a series of fast verifiers relies on back-propagating linear bounds through the network (Zhang et al., 2018; Xu et al., 2021; Wang et al., 2021). Tight bounds can be plugged in within a branch-and-bound framework (Bunel et al., 2018; 2020b) to yield a verifier that solve the verification problem in equation (2) exactly for piece-wise linear networks, with state-of-the-art complete verifiers often relying on a combination of linear bound propagation with dual bounds (Ferrari et al., 2022; Zhang et al., 2022b). Earlier certified training schemes (Gowal et al., 2018; Wong & Kolter, 2018; Zhang et al., 2020; Mirman et al., 2018; Shi et al., 2021) typically rely on linear bound propagation or IBP to create a loss function for certified training using equation (3). As described in §2.3.2, later schemes combine this with adversarial training, attaining better certified robustness when using branch-and-bound post-training. We here rely on expressive losses (§3.2) owing to their versatility and state-of-the-art performance. Finally, while we focus on ℓ_∞ certified training schemes applicable to ReLU networks and based on deterministic lower bounds to $\boldsymbol{z}_{\boldsymbol{f_\theta}}^{\mathcal{B}_\epsilon(\boldsymbol{x}),y}$, alternative techniques include randomized methods (Cohen et al., 2019), Lipschitz regularization (Huang et al., 2021), and ad-hoc architectures (Xu et al., 2022; Zhang et al., 2022a).

4.2 Adversarial Training and Catastrophic Overfitting

For simplicity, this work focuses on the widely popular multi-step PGD training (Madry et al., 2018). Nevertheless, multiple variants have been designed to improve its performance (Zhang et al., 2019; Wang et al., 2019; Liu et al., 2021; Xu et al., 2023). Multi-step PGD is coupled with random search within AutoAttack (Croce & Hein, 2020), a strong attack suite that has become the standard way to reliably evaluate empirical robustness. It was noted that adversarial training is prone to overfitting with respect to empirical robustness, a phenomenon known as robust overfitting (Rice et al., 2020), with specialized remedies including smoothing methods (Chen et al., 2020) and techniques linked to a game-theoretical interpretation of the phenomenon (Wang et al., 2024). In this work, we are rather concerned with catastrophic overfitting, a thoroughly-studied (Vivek & Babu, 2020; Park & Lee, 2021; Li et al., 2022; Tsiligkaridis & Roberts, 2022) failure mode associated with one-step adversarial training, and which can be reduced without any additional cost by the appropriate addition of noise to the attacks (Wong et al., 2020; de Jorge et al., 2022): see §2.1. Other techniques include explicit regularizers on the loss smoothness (Sriramanan et al., 2020; 2021), dynamically varying the attack step size (Kim et al., 2021), or selectively zeroing some coordinates of the attack step (Golgooni et al., 2023): these are all outperformed by the noise-based N-FGSM (de Jorge et al., 2022). Regrettably, even N-FGSM suffer from catastrophic overfitting at larger perturbation radii. Better robustness at large ϵ, albeit at additional cost, can be obtained by promoting the local linearity of the network, which is closely-linked with catastrophic overfitting (Ortiz-Jimenez et al., 2023): GradAlign (Andriushchenko & Flammarion, 2020) aligns the input gradients of clean and randomly perturbed samples, ELLE (Rocamora et al., 2024) by enforcing a simple condition on two random samples from the perturbation set and their convex combination. Recent alternatives to attain large-ϵ robustness include hindering the generation of adversarial examples with a larger loss than the clean input (Lin et al., 2023), or the use of adaptive weight perturbations (Lin et al., 2024). In section 5.1 we systematically investigate, to the best of our knowledge for the first time, whether catastrophic overfitting can be instead mitigated through certified training techniques.

4.3 Empirical Robustness of Certified Training

While it was not their main focus, previous works have reported on the empirical robustness of certified training schemes deployed to maximize certified robustness. Gowal et al. (2018) showed that IBP can attain larger empirical robustness on relatively shallow networks for MNIST with $\epsilon \in \{0.3, 0.4\}$ and for CIFAR-10 with $\epsilon = {}^{8}\!/_{255}$ when using long training schedules. A very recent work (Mao et al., 2024a) presents similar results, showing that a series of certified training schemes tuned to maximize certified accuracy outperform multi-step PGD on MNIST with $\epsilon = 0.3$ and CIFAR-10 with $\epsilon = {}^{8}\!/_{255}$. Nevertheless, given the gap between the best empirical and certified defenses across various setups (Croce & Hein, 2020; Jovanović et al., 2022; De Palma et al., 2024b), certified training schemes are commonly understood to be at odds with empirical accuracy. Expanding on the above results and complementing §5.1, section 5.2 studies whether techniques from §3, *when explicitly tuned for the purpose*, can improve empirical robustness compared to multi-step PGD on larger perturbation radii, harder datasets, and shorter training schedules. Detailed comments on the relationship between the results in §5.2 and those from Gowal et al. (2018); Mao et al. (2024a) are provided in appendix E.



(a) PGD-7 (solid) and IBP (dashed) acc., $\epsilon = {}^{8}/_{255}$.

(b) PGD-7 (solid) and IBP (dashed) acc., $\epsilon = {}^{24}/_{255}$.

(c) IBP loss for $\epsilon = {}^{8}/_{255}$.

(d) IBP loss for $\epsilon = {}^{24}/_{255}$.

Figure 5: The use of certified training techniques on top of FGSM can prevent CO for `PreActResNet18` on the CIFAR-10 test set under perturbations of $\epsilon = {}^{8}/_{255}$ and $\epsilon = {}^{24}/_{255}$.

5 Experimental Study

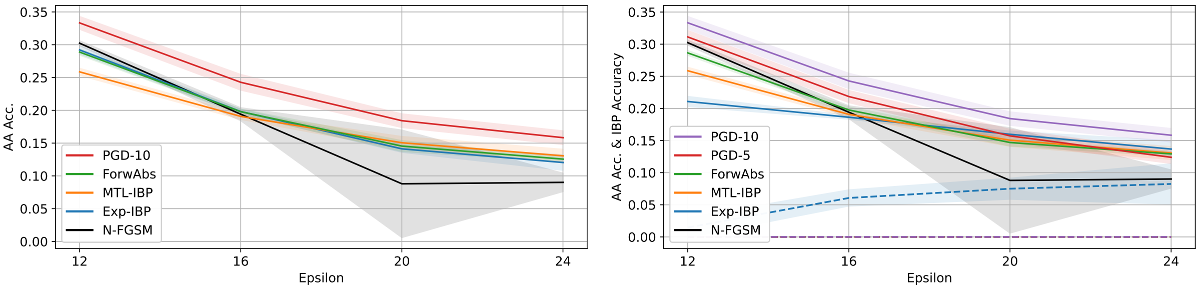
We now present the results of our experimental study on the utility of certified training schemes towards empirical robustness. Appendices B to D report omitted details and supplementary experiments.

5.1 Preventing Catastrophic Overfitting

This section experimentally demonstrates that Exp-IBP and ForwAbs can consistently prevent CO when applied on top of single-step adversarial training schemes.

FGSM The seminal work from Wong et al. (2020) established that FGSM, which is arguably the most vulnerable single-step attack, suffers from CO on CIFAR-10 from as early as $\epsilon = {}^{8}/_{255}$. Therefore, assessing whether MTL-IBP, Exp-IBP and ForwAbs can be used to prevent CO when FGSM is used to generate the adversarial point $\boldsymbol{x}_{\text{adv},\mathbb{A}}$ is a crucial qualitative test. We consider two settings: $\epsilon = {}^{8}/_{255}$, and the much harder $\epsilon = {}^{24}/_{255}$. In line with the single-step adversarial training literature (Andriushchenko & Flammarion, 2020; de Jorge et al., 2022; Rocamora et al., 2024) and section 3, we focus on `PreactResNet18`, trained via a cyclic learning rate schedule for 30 epochs. We run the three algorithms with a varying degree of regularization on their over-approximation tightness, plotting for each method the successful runs (i.e., preventing CO) having the smallest α and λ values (see appendix C.3). Figure 5 shows that, on both setups, all the three considered certified training schemes can prevent CO. For $\epsilon = {}^{8}/_{255}$, the empirical robustness of FGSM to multi-step attacks suddenly drops to near-0 roughly mid-way through training (as shown in appendix D.3, this corresponds to a sudden spike in training FGSM accuracy), and is already null at epoch 3 for $\epsilon = {}^{24}/_{255}$ (while displaying large FGSM accuracy). Differently from MTL-IBP and ForwAbs, in this setting Exp-IBP can only prevent CO while enforcing extremely low IBP loss values, resulting in non-negligible IBP accuracies on both $\epsilon = {}^{8}/_{255}$ and $\epsilon = {}^{24}/_{255}$. This is associated to a smaller empirical robustness than the other certified training methods on $\epsilon = {}^{8}/_{255}$, but not on $\epsilon = {}^{24}/_{255}$, where Exp-IBP outperforms the other algorithms also in empirical robustness. In both cases, ForwAbs attains very similar IBP losses to MTL-IBP at the end of training.

N-FGSM As explained in §2.2 and §4, the addition of noise on top of FGSM can mitigate CO at lower perturbation radii without any overhead. However, previous single-step adversarial training work (Lin et al., 2023; 2024; Rocamora et al., 2024) showed that even the state-of-the-art in noise-based single-step adversarial training, N-FGSM, becomes vulnerable when ϵ is larger. In order to complement the results of figure 5, we now consider a more realistic setting where certified training schemes rely on adversarial points $\boldsymbol{x}_{\text{adv},\mathbb{A}}$ generated through N-FGSM, investigating the empirical robustness cost of preventing CO at large ϵ values from the

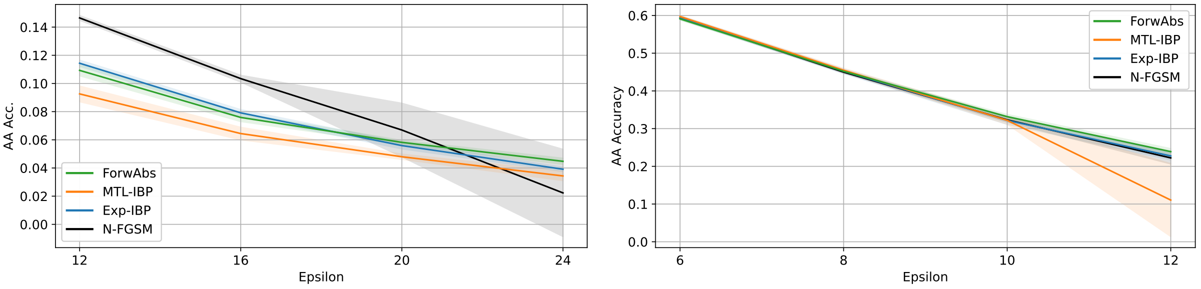


(a) AA Accuracy. Exp-IBP and ForwAbs tuned for PGD-50 accuracy.

(b) AA (solid lines) and IBP (dashed) accuracies. Exp-IBP and ForwAbs tuned for AA accuracy.

Figure 6: Certified training techniques can prevent catastrophic overfitting for N-FGSM when training `PreactResNet18` on CIFAR-10. We report means over 5 repetitions and their 95% confidence intervals.

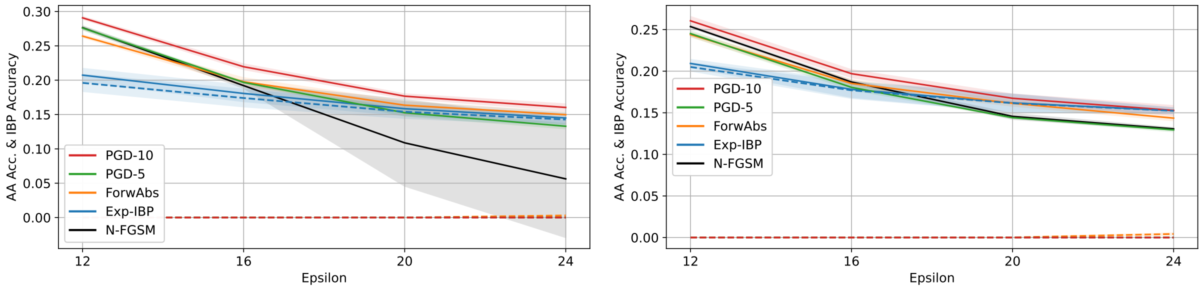
literature (Rocamora et al., 2024). As for the FGSM experiments, we use `PreactResNet18` with a 30-epoch cyclic training schedule. We tune MTL-IBP, Exp-IBP and ForwAbs to maximize empirical robustness on a holdout validation set, with $\epsilon = {}^{24}/_{255}$ for CIFAR-10 and CIFAR-100, and with $\epsilon = {}^{12}/_{255}$ for SVHN. We then retrain on the full training set with the same chosen value for all the considered perturbation radii, reporting the test-set AutoAttack (AA) accuracy (and IBP accuracy when non-null). In order to illustrate the effect of tuning on performance, we use two different tuning criteria for CIFAR-10: PGD-50 accuracy (significantly less expensive), and AA accuracy. For this network, on CIFAR-100 and SVHN, we found the single-seed tuning employed in the rest of the paper to be a poor marker of final performance. Figure 7 reports values tuned to maximize mean validation PGD-50 accuracy on 5 seeds while preventing CO. Figure 6 shows that all the three considered algorithms can prevent CO for CIFAR-10, resulting in significantly larger robustness than N-FGSM on both $\epsilon = {}^{20}/_{255}$ and $\epsilon = {}^{24}/_{255}$. While the AA-tuned runs pay a large price in empirical robustness for $\epsilon = {}^{12}/_{255}$, they remarkably attain stronger AA robustness than PGD-5 for $\epsilon = {}^{24}/_{255}$. Furthermore, for $\epsilon \geq {}^{20}/_{255}$ Exp-IBP yields an IBP accuracy comparable to the average AA accuracy of N-FGSM. As shown in Figure 7, CIFAR-100 and SVHN display instead worse trade-offs. While MTL-IBP, Exp-IBP and ForwAbs all prevent CO on CIFAR-100 for $\epsilon \geq {}^{20}/_{255}$, as visible from the reduced confidence intervals, they do so at a large cost in average empirical robustness for $\epsilon < {}^{24}/_{255}$. On the SVHN experiments, N-FGSM never suffers from CO. ForwAbs results in relatively small yet consistent performance improvements, while improvements from Exp-IBP are negligible. Remarkably, MTL-IBP fails to drive the IBP loss sufficiently low (cf. figure 19d) and appears to be more vulnerable to CO than N-FGSM at $\epsilon = {}^{12}/_{255}$. We ascribe this to the extreme sensitivity of MTL-IBP in the considered setup, linked to the qualitative properties of its loss on deeper networks described in §3.2 (see figure 3). We exclude MTL-IBP from the rest of the empirical study owing to this failure case. While the primary goal of this work is to study the utility of certified training schemes for empirical robustness (as opposed to outperforming the state-of-the-art in single-step adversarial training), a comparison with ELLE (Rocamora et al., 2024), a recent state-of-the-art regularizer, is presented in ap-



(a) CIFAR-100

(b) SVHN

Figure 7: Effect of certified training techniques on catastrophic overfitting when training `PreactResNet18` on CIFAR-100 and SVHN. We report means over 5 repetitions and their 95% confidence intervals.



(a) `CNN-7`.

(b) `CNN-5`.

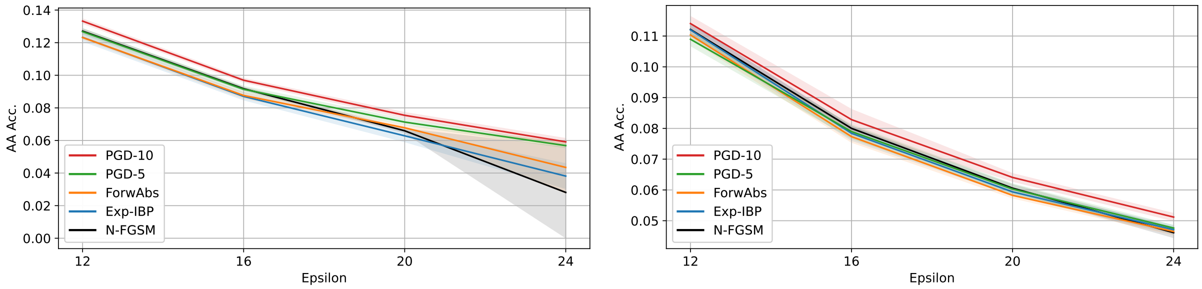
Figure 8: `CNN` comparison of certified training techniques with adversarial training for varying ϵ on CIFAR10. AutoAttack (solid lines) and IBP (dashed) accuracies are reported (means and 95% CIs for 5 runs).

pendix D.5 for reference purposes. On CIFAR-10, MTL-IBP, Exp-IBP and ForwAbs all match or outperform ELLE in terms of AA accuracy for $\epsilon \geq {}^{20}/_{255}$, highlighting their particularly strong performance on easier datasets. On the other hand, on CIFAR-100 ELLE significantly outperforms certified training techniques at all perturbation radii. We refer the reader to appendix D.5 for a more detailed discussion. In conclusion of this section, we point out that the empirical robustness costs on lower perturbation radii can be mitigated by separately tuning α and λ on them (for instance, the performance of N-FGSM can be trivially preserved via $\alpha = \lambda = 0$). Nevertheless, optimizing performance on lower ϵ is out of the scope of this experiment.

5.2 Bridging the Gap to Multi-Step Adversarial Training

We here demonstrate that the gap between certified training techniques based on single-step attacks and multi-step baselines can be reduced by training on shallower networks and with longer schedules, in some settings outperforming PGD-10 in empirical robustness without employing an adversarial training component.

Cyclic training schedule We replicate the cyclic-schedule experiments from figures 6 and 7 on two shallower networks without skip connections. As in figures 6 and 7, $\boldsymbol{x}_{\text{adv},\mathbb{A}}$ for certified training methods is generated via N-FGSM. First, we consider `CNN-7`, a 7-layer convolutional network from the certified training literature (Shi et al., 2021; Müller et al., 2023; Mao et al., 2023; De Palma et al., 2024b). Then, to study the impact of network width, we also provide results on a 5-layer version, named `CNN-5` (see appendix C.3.1). We tune Exp-IBP and ForwAbs to maximize AA accuracy for $\epsilon = {}^{24}/_{255}$ on a validation set for each dataset and network, using the tuned value for all considered ϵ. Figure 8 demonstrates that both ForwAbs and Exp-IBP outperform PGD-5 in empirical robustness for $\epsilon \geq {}^{20}/_{255}$ on both `CNN` models, with ForwAbs preserving AA accuracy at lower epsilons, and Exp-IBP attaining significant IBP accuracy (at least as large as PGD-5's AA accuracy for $\epsilon \geq {}^{20}/_{255}$). The gap to PGD-10 is also significantly reduced compared to the `PreActResNet18` experiments, in spite of the significant CO shown by N-FGSM, the attack employed within both Exp-IBP and



(a) `CNN-7`.

(b) `CNN-5`.

Figure 9: `CNN` comparison of certified training techniques with adversarial training for varying ϵ on CIFAR100. AutoAttack (solid lines) and IBP (dashed) accuracies are reported (means and 95% CIs for 5 runs).

ForwAbs. Furthermore, as shown in appendix D.2, both Exp-IBP and ForwAbs reduce runtime compared to PGD-5 and PGD-10. Remarkably, on `CNN-5`, as shown in figure 8b, Exp-IBP matches the performance of PGD-10 for $\epsilon = 24/255$, significantly outperforming N-FGSM, which does not suffer from CO in this setup. In order to prevent ForwAbs from under-performing on `CNN-5`, we found it crucial to include sufficiently low λ values in the tuning grid: see appendix C.3.3. Finally, figure 9 shows that, while the overall performance trade-offs are greatly improved compared to `PreActResNet18`, Exp-IBP and ForwAbs fail to improve on multi-step attacks on CIFAR-100 also on the `CNN` architectures. We ascribe this to the larger number of classes, which produce a less favorable trade-off between empirical and certified robustness.

Long training schedule We now investigate whether certified training schemes can be more beneficial under longer training schedules. We consider a schedule of 160 epochs from the certified training literature (Shi et al., 2021) in which the learning rate is decayed twice towards the end of training, and the perturbation radius (for all methods and loss components) is gradually increased from 0 to its target value during the first 80 epochs. In order to show the benefits of tuning for each ϵ, the hyper-parameters of Exp-IBP and ForwAbs (α and λ, respectively) are tuned to maximize validation AA accuracy individually on each setup considered in table 1, with the exception of $\epsilon = 16/255$ CIFAR-10, which re-uses the values from $\epsilon = 24/255$ to reduce computational overhead. As in figures 6b to 9, N-FGSM is used to generate $\boldsymbol{x}_{\text{adv},\mathbb{A}}$ for Exp-IBP and ForwAbs, except for "Exp-IBP PGD-5", which relies on PGD-5 instead. Noticing the beneficial effect of large α values, we also include pure IBP ($\alpha = 1$), whose loss is attack-free, in the comparison. Table 1 shows that Exp-IBP outperforms PGD-5 at all considered CIFAR-10 setups. While empirical robustness is maximized when the IBP accuracy is large on $\epsilon = 24/255$, it is not the case for $\epsilon = 8/255$. Remarkably, pure IBP training outperforms all other methods for $\epsilon \geq 16/255$ on CIFAR-10. Exp-IBP PGD-5 demonstrates that on CIFAR-10 certified training techniques can also improve robustness when applied on top of multi-step attacks: this appears to be useful to improve performance when single-step attacks systematically fail. While ForwAbs can significantly boost the empirical robustness of N-FGSM, it manages to outperform PGD-5 only on CIFAR-10 with $\epsilon = 24/255$. Finally, the long schedule does not benefit certified training schemes on CIFAR-100, where their relative performance compared to multi-step attacks remains unvaried compared to the cyclic schedule, and for which we did not find the use of PGD-5 to generate the attacks for Exp-IBP to be beneficial.

Table 1: Comparison for the long training schedule on `CNN-7`. Bold entries indicate the best AA or IBP accuracy for each setting. Italics denote AA accuracy improvements on PGD-5.

Dataset	ϵ	Method	AA acc. [%]	IBP acc. [%]
CIFAR-10	$\frac{8}{255}$	Exp-IBP	*38.44 ± 0.25*	0.00
		IBP	29.43 ± 0.96	**29.02 ± 0.96**
		ForwAbs	*37.92 ± 0.08*	0.00
		PGD-10	**40.14 ± 0.65**	0.00
		PGD-5	37.32 ± 0.71	0.00
		N-FGSM	37.76 ± 0.30	0.00
	$\frac{16}{255}$	Exp-IBP	*20.72 ± 1.54*	20.55 ± 1.45
		IBP	**21.72 ± 0.68**	**21.62 ± 0.68**
		ForwAbs	17.61 ± 0.38	0.00
		PGD-10	21.13 ± 0.60	0.00
		PGD-5	18.19 ± 0.32	0.00
		N-FGSM	7.13 ± 11.43	0.00
	$\frac{24}{255}$	Exp-IBP	*14.79 ± 1.61*	14.74 ± 1.59
		Exp-IBP PGD-5	*16.45 ± 1.32*	16.34 ± 1.29
		IBP	**16.73 ± 0.43**	**16.56 ± 0.42**
		ForwAbs	*14.39 ± 0.16*	0.46 ± 0.17
		PGD-10	*15.14 ± 0.46*	0.00
		PGD-5	11.47 ± 0.29	0.00
		N-FGSM	0.23 ± 0.64	0.00
CIFAR-100	$\frac{24}{255}$	Exp-IBP	3.25 ± 0.44	3.14 ± 0.41
		IBP	3.78 ± 0.16	**3.67 ± 0.16**
		ForwAbs	4.31 ± 0.05	0.00
		PGD-10	**6.15 ± 0.16**	0.00
		PGD-5	5.52 ± 0.18	0.00
		N-FGSM	0.02 ± 0.06	0.00

6 Conclusions

We presented a comprehensive empirical study on the utility of recent certified training techniques for empirical robustness, as opposed to verifiability, their original design goal. In particular, we showed that Exp-IBP can prevent catastrophic overfitting on single-step attacks for a variety of settings, outperforming some multi-step baselines in a subset of these. Furthermore, we introduced a novel regularizer on the size of network over-approximations, named ForwAbs, that can achieve similar effects to Exp-IBP while cutting down its overhead on top of single-step attacks. While we believe that these results highlight the potential of certified training as an empirical defense, they also show the severe limitations of current techniques on harder datasets, ultimately calling for the development of better certified training algorithms.

References

Ross Anderson, Joey Huchette, Will Ma, Christian Tjandraatmadja, and Juan Pablo Vielma. Strong mixed-integer programming formulations for trained neural networks. *Mathematical Programming*, 2020.

Maksym Andriushchenko and Nicolas Flammarion. Understanding and improving fast adversarial training. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 16048–16059. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/b8ce47761ed7b3b6f48b583350b7f9e4-Paper.pdf`.

Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *Computer Vision – ECCV 2020*, pp. 484–501, Cham, 2020. Springer International Publishing.

Mislav Balunovic and Martin Vechev. Adversarial training and provable defenses: Bridging the gap. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=SJxSDxrKDr`.

Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*. Springer, 2013.

Rudy Bunel, Ilker Turkaslan, Philip HS Torr, Pushmeet Kohli, and M Pawan Kumar. A unified view of piecewise linear neural network verification. *Neural Information Processing Systems*, 2018.

Rudy Bunel, Alessandro De Palma, Alban Desmaison, Krishnamurthy Dvijotham, Pushmeet Kohli, Philip HS Torr, and M Pawan Kumar. Lagrangian decomposition for neural network verification. *Conference on Uncertainty in Artificial Intelligence*, 2020a.

Rudy Bunel, Jingyue Lu, Ilker Turkaslan, P Kohli, P Torr, and M Pawan Kumar. Branch and bound for piecewise linear neural network verification. *Journal of Machine Learning Research*, 21(2020), 2020b.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57, 2017. doi: 10.1109/SP.2017.49.

Tianlong Chen, Zhenyu Zhang, Sijia Liu, Shiyu Chang, and Zhangyang Wang. Robust overfitting may be mitigated by properly learned smoothening. In *International Conference on Learning Representations*, 2020.

Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *International conference on machine learning*, 2017.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, 2019.

Patrick Cousot and Radhia Cousot. Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In Robert M. Graham, Michael A. Harrison, and Ravi Sethi (eds.), *Conference Record of the Fourth ACM Symposium on Principles of Programming Languages, Los Angeles, California, USA, January 1977*, pp. 238–252. ACM, 1977. doi: 10.1145/512950.512973. URL `https://doi.org/10.1145/512950.512973`.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020.

Pau de Jorge, Adel Bibi, Riccardo Volpi, Amartya Sanyal, Philip Torr, Grégory Rogez, and Puneet K. Dokania. Make some noise: Reliable and efficient single-step adversarial training. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=NENo__bExYu`.

Alessandro De Palma, Harkirat Singh Behl, Rudy Bunel, Philip H. S. Torr, and M. Pawan Kumar. Scaling the convex barrier with active sets. *International Conference on Learning Representations*, 2021.

Alessandro De Palma, Rudy Bunel, Krishnamurthy Dvijotham, M. Pawan Kumar, and Robert Stanforth. IBP regularization for verified adversarial robustness via branch-and-bound. In *ICML 2022 Workshop on Formal Verification of Machine Learning*, 2022.

Alessandro De Palma, Harkirat Singh Behl, Rudy Bunel, Philip H. S. Torr, and M. Pawan Kumar. Scaling the convex barrier with sparse dual algorithms. *Journal of Machine Learning Research*, 2024a.

Alessandro De Palma, Rudy Bunel, Krishnamurthy Dvijotham, M. Pawan Kumar, Robert Stanforth, and Alessio Lomuscio. Expressive losses for verified robustness via convex combinations. In *The Twelfth International Conference on Learning Representations*, 2024b. URL `https://openreview.net/forum?id=mzyZ4wzKlM`.

Krishnamurthy Dvijotham, Robert Stanforth, Sven Gowal, Timothy Mann, and Pushmeet Kohli. A dual approach to scalable verification of deep networks. *Conference on Uncertainty in Artificial Intelligence*, 2018.

Ruediger Ehlers. Formal verification of piece-wise linear feed-forward neural networks. *Automated Technology for Verification and Analysis*, 2017.

Jiameng Fan and Wenchao Li. Adversarial training and provable robustness: A tale of two objectives. In *AAAI Conference on Artificial Intelligence*, 2021.

Claudio Ferrari, Mark Niklas Mueller, Nikola Jovanović, and Martin Vechev. Complete verification via multi-neuron relaxation guided branch-and-bound. *International Conference on Learning Representations*, 2022.

Zeinab Golgooni, Mehrdad Saberi, Masih Eskandar, and Mohammad Hossein Rohban. Zerograd: Costless conscious remedies for catastrophic overfitting in the fgsm adversarial training. *Intelligent Systems with Applications*, 19:200258, 2023. ISSN 2667-3053. doi: https://doi.org/10.1016/j.iswa.2023.200258. URL `https://www.sciencedirect.com/science/article/pii/S2667305323000832`.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL `http://arxiv.org/abs/1412.6572`.

Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:1810.12715*, 2018. URL `https://arxiv.org/abs/1810.12715`.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pp. 630–645. Springer, 2016.

P. Henriksen and A. Lomuscio. Deepsplit: An efficient splitting method for neural network verification via indirect effect analysis. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI21)*, 2021.

Yujia Huang, Huan Zhang, Yuanyuan Shi, J Zico Kolter, and Anima Anandkumar. Training certifiably robust neural networks with efficient local lipschitz bounds. In *Neural Information Processing Systems*, 2021.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, 2015.

Nikola Jovanović, Mislav Balunovic, Maximilian Baader, and Martin Vechev. On the paradox of certified training. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=atJHLVyBi8.

Guy Katz, Clark Barrett, David Dill, Kyle Julian, and Mykel Kochenderfer. Reluplex: An efficient SMT solver for verifying deep neural networks. *arXiv preprint arXiv:1702.01135*, 2017.

Hoki Kim, Woojin Lee, and Jaewook Lee. Understanding catastrophic overfitting in single-step adversarial training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 8119–8127, 2021.

A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

Tao Li, Yingwen Wu, Sizhe Chen, Kun Fang, and Xiaolin Huang. Subspace adversarial training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Runqi Lin, Chaojian Yu, and Tongliang Liu. Eliminating catastrophic overfitting via abnormal adversarial examples regularization. *Advances in Neural Information Processing Systems*, 2023.

Runqi Lin, Chaojian Yu, Bo Han, Hang Su, and Tongliang Liu. Layer-aware analysis of catastrophic overfitting: Revealing the pseudo-robust shortcut dependency. *International Conference on Machine Learning*, 2024.

Feng Liu, Bo Han, Tongliang Liu, Chen Gong, Gang Niu, Mingyuan Zhou, Masashi Sugiyama, et al. Probabilistic margins for instance reweighting in adversarial training. *Advances in Neural Information Processing Systems*, 2021.

Alessio Lomuscio and Lalit Maganti. An approach to reachability analysis for feed-forward ReLU neural networks. *arXiv:1706.07351*, 2017.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJzIBfZAb.

Yuhao Mao, Mark Niklas Mueller, Marc Fischer, and Martin Vechev. Connecting certified and adversarial training. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=T2lM4ohRwb.

Yuhao Mao, Stefan Balauca, and Martin Vechev. CTBENCH: A library and benchmark for certified training. *arXiv preprint arXiv:2406.04848*, 2024a.

Yuhao Mao, Mark Niklas Müller, Marc Fischer, and Martin Vechev. Understanding certified training with interval bound propagation. *International Conference on Learning Representations*, 2024b.

Matthew Mirman, Timon Gehr, and Martin Vechev. Differentiable abstract interpretation for provably robust neural networks. In *International Conference on Machine Learning*, 2018.

R.E. Moore. *Interval Analysis*. Prentice-Hall series in automatic computation. Prentice-Hall, 1966. URL https://books.google.fr/books?id=csQ-AAAAIAAJ.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Mark Niklas Müller, Franziska Eckert, Marc Fischer, and Martin Vechev. Certified training: Small boxes are all you need. In *International Conference on Learning Representations*, 2023.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. URL `http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf`.

Guillermo Ortiz-Jimenez, Pau de Jorge, Amartya Sanyal, Adel Bibi, Puneet K Dokania, Pascal Frossard, Grégory Rogez, and Philip Torr. Catastrophic overfitting can be induced with discriminative non-robust features. *Transactions on Machine Learning Research*, 2023.

Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *IEEE symposium on security and privacy (SP)*, 2016.

Geon Yeong Park and Sang Wan Lee. Reliably fast adversarial training via latent adversarial perturbation. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2021.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. *Neural Information Processing Systems*, 2019.

Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *International conference on machine learning*, pp. 8093–8104. PMLR, 2020.

Elias Abad Rocamora, Fanghui Liu, Grigorios Chrysos, Pablo M. Olmos, and Volkan Cevher. Efficient local linearity regularization to overcome catastrophic overfitting. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=SZzQz8ikwg`.

Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper_files/paper/2019/file/7503cfacd12053d309b6bed5c89de212-Paper.pdf`.

Zhouxing Shi, Yihan Wang, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. Fast certified robust training with short warmup. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 18335–18349. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper_files/paper/2021/file/988f9153ac4fd966ea302dd9ab9bae15-Paper.pdf`.

Gagandeep Singh, Timon Gehr, Matthew Mirman, Markus Püschel, and Martin Vechev. Fast and effective robustness certification. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 31*, pp. 10802–10813. Curran Associates, Inc., 2018. URL `http://papers.nips.cc/paper/8278-fast-and-effective-robustness-certification.pdf`.

Gagandeep Singh, Rupanshu Ganvir, Markus Püschel, and Martin Vechev. Beyond the single neuron convex barrier for neural network certification. *Neural Information Processing Systems*, 2019a.

Gagandeep Singh, Timon Gehr, Markus Püschel, and Martin Vechev. An abstract domain for certifying neural networks. *Proceedings of the ACM on Programming Languages*, 3(POPL):1–30, 2019b. doi: 10.1145/3290354. URL `https://doi.org/10.1145/3290354`.

Gaurang Sriramanan, Sravanti Addepalli, Arya Baburaj, and Venkatesh Babu R. Guided adversarial attack for evaluating and enhancing adversarial defenses. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 20297–20308. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/ea3ed20b6b101a09085ef09c97da1597-Paper.pdf`.

Gaurang Sriramanan, Sravanti Addepalli, Arya Baburaj, and Venkatesh Babu Radhakrishnan. Towards efficient and effective adversarial training. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=kuK2VARZGnI.

Teruo Sunaga. Theory of an interval algebra and its application to numerical analysis. *RAAG Memoirs*, 1958.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In Yoshua Bengio and Yann LeCun (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6199.

Christian Tjandraatmadja, Ross Anderson, Joey Huchette, Will Ma, Krunal Patel, and Juan Pablo Vielma. The convex relaxation barrier, revisited: Tightened single-neuron relaxations for neural network verification. *Neural Information Processing Systems*, 2020.

Vincent Tjeng, Kai Xiao, and Russ Tedrake. Evaluating Robustness of Neural Networks with Mixed Integer Programming. 2019. URL https://openreview.net/pdf?id=HyGIdiRqtm.

Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *International Conference on Learning Representations*, 2018.

Theodoros Tsiligkaridis and Jay Roberts. Understanding and increasing efficiency of frank-wolfe adversarial training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

BS Vivek and R Venkatesh Babu. Single-step adversarial training with dropout scheduling. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020.

Shiqi Wang, Huan Zhang, Kaidi Xu, Xue Lin, Suman Jana, Cho-Jui Hsieh, and J Zico Kolter. Beta-CROWN: Efficient bound propagation with per-neuron split constraints for complete and incomplete neural network verification. *Neural Information Processing Systems*, 2021.

Yifei Wang, Liangchen Li, Jiansheng Yang, Zhouchen Lin, and Yisen Wang. Balance, imbalance, and rebalance: Understanding robust overfitting from a minimax game perspective. *Advances in neural information processing systems*, 36, 2024.

Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International conference on learning representations*, 2019.

Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, 2018.

Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BJx040EFvH.

Kaidi Xu, Zhouxing Shi, Huan Zhang, Yihan Wang, Kai-Wei Chang, Minlie Huang, Bhavya Kailkhura, Xue Lin, and Cho-Jui Hsieh. Automatic perturbation analysis for scalable certified robustness and beyond. In *Neural Information Processing Systems*, 2020.

Kaidi Xu, Huan Zhang, Shiqi Wang, Yihan Wang, Suman Jana, Xue Lin, and Cho-Jui Hsieh. Fast and complete: Enabling complete neural network verification with rapid and massively parallel incomplete verifiers. In *International Conference on Learning Representations*, 2021.

Xiaojun Xu, Linyi Li, and Bo Li. Lot: Layer-wise orthogonal training on improving l2 certified robustness. In *Neural Information Processing Systems*, 2022.

Yuancheng Xu, Yanchao Sun, Micah Goldblum, Tom Goldstein, and Furong Huang. Exploring and exploiting decision boundary dynamics for adversarial robustness. *International Conference on Learning Representations*, 2023.

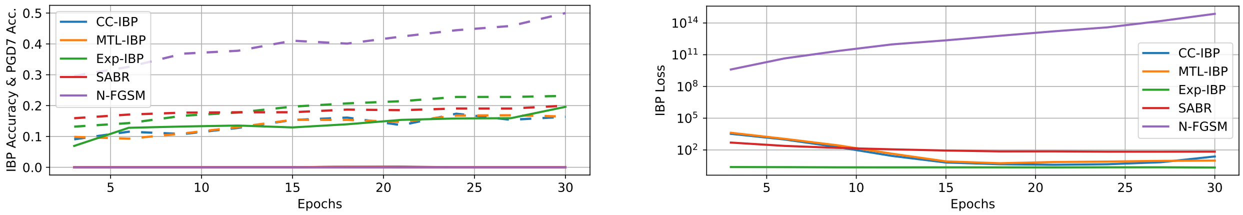
Bohang Zhang, Du Jiang, Di He, and Liwei Wang. Rethinking lipschitz neural networks and certified robustness: A boolean function perspective. In *Neural Information Processing Systems*, 2022a.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, 2019.

Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL `https://proceedings.neurips.cc/paper_files/paper/2018/file/d04863f100d59b3eb688a11f95b0ae60-Paper.pdf`.

Huan Zhang, Hongge Chen, Chaowei Xiao, Sven Gowal, Robert Stanforth, Bo Li, Duane Boning, and Cho-Jui Hsieh. Towards stable and efficient training of verifiably robust neural networks. In *International Conference on Learning Representations*, 2020.

Huan Zhang, Shiqi Wang, Kaidi Xu, Linyi Li, Bo Li, Suman Jana, Cho-Jui Hsieh, and J Zico Kolter. General cutting planes for bound-propagation-based neural network verification. *Neural Information Processing Systems*, 2022b.



(a) Certified accuracy via IBP bounds (solid lines), empirical accuracy under PGD-7 attacks (dashed).

(b) IBP loss.

Figure 10: IBP certified robustness attained by expressive losses on the `PreActResNet18` training setup from de Jorge et al. (2022). Validation results on CIFAR-10 under perturbations of $\epsilon = 8/255$.

A Qualitative Behavior of CC-IBP and SABR

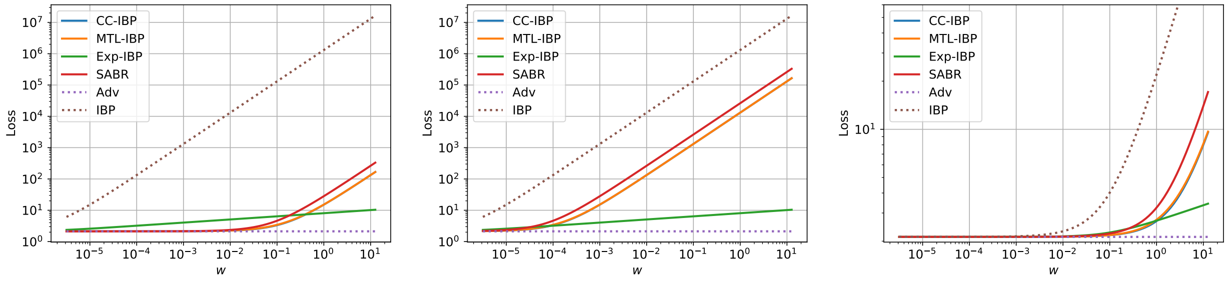
We here present the CC-IBP and SABR losses, then expand the qualitative analysis on the behavior of expressive losses presented in §3.2. Specifically, focusing again on a `PreActResNet18` setup prevalent in the single-step adversarial training literature and studied in §5.1, we will show that they share the same qualitative behavior of MTL-IBP.

Let $\boldsymbol{x}_{\text{adv},\mathbb{A}}$ be the output of the chosen attack for perturbation set $\mathcal{B}_\epsilon(\boldsymbol{x})$. SABR (Müller et al., 2023) computes the IBP loss over a subset of the perturbation set of radius $\alpha\epsilon$ centered on $\boldsymbol{x}_\alpha = \text{Proj}(\boldsymbol{x}_{\text{adv},\mathbb{A}}, \mathcal{B}_{\epsilon-\alpha\epsilon}(\boldsymbol{x}))$. Similarly to MTL-IBP and Exp-IBP (see §3.2), the CC-IBP loss is defined through convex combinations. Specifically, it computes the chosen loss $\mathcal{L}$ on the convex combination between the adversarial logit differences $\boldsymbol{z}_{\boldsymbol{f_\theta}}(\boldsymbol{x}_{\text{adv},\mathbb{A}}, y)$ and the IBP lower bounds $\boldsymbol{l}_{\boldsymbol{f_\theta}}^{\mathcal{B}_\epsilon(\boldsymbol{x}),y}$.

Definition A.1 *The SABR and CC-IBP losses are defined as follows:*

$$\begin{aligned} \mathcal{L}_\alpha^{SABR}(\boldsymbol{f}_\theta, \mathcal{B}_\epsilon(\boldsymbol{x}); y) &:= \mathcal{L}_{IBP}(\boldsymbol{f}_\theta, \mathcal{B}_{\alpha\epsilon}(\boldsymbol{x}_\alpha); y), \\ \mathcal{L}_\alpha^{CC\text{-}IBP}(\boldsymbol{f}_\theta, \mathcal{B}_\epsilon(\boldsymbol{x}); y) &:= \mathcal{L}\left(-\left[(1-\alpha)\boldsymbol{z}_{\boldsymbol{f_\theta}}(\boldsymbol{x}_{adv,\mathbb{A}}, y) + \alpha\, \boldsymbol{l}_{\boldsymbol{f_\theta}}^{\mathcal{B}_\epsilon(\boldsymbol{x}),y}\right], y\right), \end{aligned} \tag{7}$$

Figure 10 shows that, as MTL-IBP, SABR and CC-IBP fail to yield non-negligible certified accuracy via IBP. As for MTL-IBP, we gradually increase α and the IBP perturbation radius to improve performance (see appendix D.1). Their final IBP losses are both larger than those attained by MTL-IBP (and, transitively, by Exp-IBP). In addition, figure 11 shows that CC-IBP and SABR display the same growth trends of the MTL-IBP loss on the toy networks, further demonstrating their common similarity.



(a) 18 layers; $\alpha = 10^{-5}$ for CC-IBP, MTL-IBP and SABR.

(b) 18 layers; $\alpha = 10^{-2}$ for CC-IBP, MTL-IBP and SABR.

(c) 2 layers; $\alpha = 3 \times 10^{-2}$ for CC-IBP, MTL-IBP and SABR.

Figure 11: Sensitivity of the expressive losses on a toy network of varying depth, with $\alpha = 10^{-1}$ for Exp-IBP. For all three plots, CC-IBP displays almost identical behavior to MTL-IBP.

B Experimental Details

We now provide experimental details omitted from sections 3 to 5.2.

B.1 Toy Networks

The toy neural networks employed in §3.2 are fully connected networks of depth n designed to feature IBP bounds which are tunable in magnitude through a scalar parameter w, and explode with the depth of the network. They are defined as follows:

$$\boldsymbol{x}^1 = \text{ReLU}\left(\begin{bmatrix} w & -w \\ -w & w \end{bmatrix} \boldsymbol{x}^0\right), \ \boldsymbol{x}^n = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \boldsymbol{x}^{n-1} + \begin{bmatrix} 3 \\ 1 \end{bmatrix}, \ \left\{\boldsymbol{x}^k = \text{ReLU}\left(\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \boldsymbol{x}^{k-1}\right)\right\} \forall \ k \in [\![2, n-1]\!],$$

evaluating to:

$$\boldsymbol{x}^n = 2^{n-1} \ \text{ReLU}\left(w \begin{bmatrix} \boldsymbol{x}^0_0 - \boldsymbol{x}^0_1 \\ \boldsymbol{x}^0_1 - \boldsymbol{x}^0_0 \end{bmatrix}\right) + \begin{bmatrix} 3 \\ 1 \end{bmatrix}.$$

Let us assume that $y = 1$ (a similar reasoning holds if $y = 0$). In this context, if $w \geq 0$, the logit differences $\boldsymbol{z}_{\boldsymbol{x}^n}(\boldsymbol{x}^0, y)$ can be computed as:

$$\begin{aligned} \boldsymbol{z}_{\boldsymbol{x}^n}(\boldsymbol{x}^0, 1) &= 2^{n-1} \left[\text{ReLU}\left(w \left(\boldsymbol{x}^0_1 - \boldsymbol{x}^0_0\right)\right) - \text{ReLU}\left(w \left(\boldsymbol{x}^0_0 - \boldsymbol{x}^0_1\right)\right)\right] - 2 \\ &= w \ 2^{n-1} \left[\text{ReLU}\left(\boldsymbol{x}^0_1 - \boldsymbol{x}^0_0\right) - \text{ReLU}\left(\boldsymbol{x}^0_0 - \boldsymbol{x}^0_1\right)\right] - 2 \\ &= w \ 2^{n-1} \left(\boldsymbol{x}^0_1 - \boldsymbol{x}^0_0\right) - 2. \end{aligned}$$

If an $\tilde{\boldsymbol{x}}^0$ such that $\tilde{\boldsymbol{x}}^0_1 - \tilde{\boldsymbol{x}}^0_0 < 0$ belongs to the perturbation set $\mathcal{B}_\epsilon(\boldsymbol{x}^0)$, then $\boldsymbol{z}_{\boldsymbol{x}^n}(\tilde{\boldsymbol{x}}^0, 1)$ will have $(-\infty, -2]$ as image for $w \in [0, +\infty)$. Noting that the relative IBP lower bounds $\boldsymbol{l}_{\boldsymbol{x}^n}^{\mathcal{B}_\epsilon(\boldsymbol{x}^0),1}$ evaluate to -2 if $w = 0$ (see equation 4), that $\boldsymbol{l}_{\boldsymbol{x}^n}^{\mathcal{B}_\epsilon(\boldsymbol{x}^0),1} \leq \boldsymbol{z}_{\boldsymbol{x}^n}(\tilde{\boldsymbol{x}}^0, 1)$ for any $w \geq 0$, and that $\boldsymbol{l}_{\boldsymbol{x}^n}^{\mathcal{B}_\epsilon(\boldsymbol{x}^0),1}$ is a continuous function with respect to w (as it is computed through compositions and linear combinations of continuous functions), the image of $\boldsymbol{l}_{\boldsymbol{x}^n}^{\mathcal{B}_\epsilon(\boldsymbol{x}^0),1}$ for $w \in [0, +\infty)$ will also be $(-\infty, -2]$. In addition, for any fixed $w > 0$, $\boldsymbol{l}_{\boldsymbol{x}^n}^{\mathcal{B}_\epsilon(\boldsymbol{x}^0),1}$ will decrease at least as fast as $\boldsymbol{z}_{\boldsymbol{x}^n}(\tilde{\boldsymbol{x}}^0, 1)$ (which decreases exponentially) with the depth of the network.

In order to meet the above condition, figure 3 uses $\boldsymbol{x}^0 = [-5, 5]^T$, $y = 1$ and $\mathcal{B}_{10}(\boldsymbol{x}^0)$ as perturbation. Furthermore, in order to keep the adversarial loss constant with w for the purposes of figure 3, we set $\boldsymbol{x}_{\text{adv},\mathbb{A}} = [0, 0]^T$, for which $\boldsymbol{z}_{\boldsymbol{x}^n}(\boldsymbol{x}_{\text{adv},\mathbb{A}}, 1) = -2$ regardless of w.

B.2 Datasets

We focus on three standard 32 × 32 image classification datasets: CIFAR-10 and CIFAR-100 (Krizhevsky & Hinton, 2009), and SVHN (Netzer et al., 2011). CIFAR-10 and CIFAR-100 consist of 60,000 32 × 32 RGB images, with 50,000 images for training and 10,000 for testing. CIFAR-10 contains 10 classes, while CIFAR-100 contains 100 classes. SVHN consists of 73,257 32 × 32 RGB images, with 73,257 images for training and 26,032 for testing.

Unless specified otherwise, for tuning purposes or when reporting validation results we use a random 20% holdout of the training set as validation set, and train on the remaining 80%. After tuning and when reporting test set results, we use the standard train and test splits for all datasets.

C Implementation Details and Computational Setup

This appendix details our implementation and outlines the computational setup employed to run the experiments.

C.1 Implementation Details

Our implementation relies on PyTorch (Paszke et al., 2019) and on the public codebases from de Jorge et al. (2022); Rocamora et al. (2024); De Palma et al. (2024b). We compute IBP bounds using the `auto_LiRPA` implementation (Xu et al., 2020).

N-FGSM may return a point outside the set of allowed perturbations $\mathcal{B}_{\epsilon}(\boldsymbol{x})$. As a result, when using it to compute the adversarial point $\boldsymbol{x}_{\text{adv},\mathbb{A}}$ for SABR, we disable the projection of the SABR perturbation subset onto the original perturbation set, simply setting $\boldsymbol{x}_{\alpha} = \boldsymbol{x}_{\text{adv},\mathbb{A}}$ (see §3.2), which allows the implementation to meet the definition of expressivity for small α values.

Consistently with the single-step adversarial training literature (de Jorge et al., 2022; Rocamora et al., 2024), BatchNorm layers (Ioffe & Szegedy, 2015), which are present in all the employed models, are always kept in training mode throughout training (including during the attacks). When computing IBP bounds for expressive losses and ForwAbs terms at training time, we use the batch statistics from the current adversarial attack. For ForwAbs, these are retrieved from the `auto_LiRPA` implementation (Xu et al., 2020). The clean inputs are never fed to the network at training time, except when using FGSM to generate the attack or when training with pure IBP (in that case, the clean batch statistics are used to compute the IBP bounds). The outcome of the running statistics compute during training is systematically employed at evaluation time.

C.2 Computational Setup

All timing measurements were carried out on an Nvidia GTX 1080Ti GPU, using 6 cores of an Intel Skylake Xeon 5118 CPU. All the other experiments were run on a single GPU each, allocated from two separate Slurm-based internal clusters. We used the following GPU models from one cluster: Nvidia V100, Nvidia RTX6000, Nvidia RTX8000, Nvidia GTX 1080Ti, Nvidia RTX2080Ti. And the following GPU models from the other cluster: Nvidia Quadro P5000, Nvidia H100.

C.3 Training Details and Hyper-parameters

We now provide omitted training details and hyper-parameters.

C.3.1 Network Architectures

The `PreActResNet18` and `CNN-7` architectures used in our experiments are left unvaried with respect to the implementations from de Jorge et al. (2022) and De Palma et al. (2024b), respectively.

The `CNN-5` architecture has the following structure:

1. convolutional layer with 64 3×3 filters, stride = 1 and padding = 1, followed by BatchNorm and a ReLU activation function;
2. convolutional layer with 64 4×4 filters, stride = 2 and padding = 1, then BatchNorm and ReLU;
3. convolutional layer with 128 4×4 filters, stride = 2 and padding = 1, then BatchNorm and ReLU;
4. linear layer with 512 neurons, then BatchNorm and ReLU;
5. linear layer with k (the number of classes) neurons.

C.3.2 Initialization and Training Schedule

All the networks trained using MTL-IBP, Exp-IBP and ForwAbs are initialized using the specialized technique from Shi et al. (2021), which results in smaller IBP bounds at initialization. Except for the experiments from §3 and for those in table 1, where the specialized initialization is employed in order to fairly compare IBP bounds, all the networks trained via pure adversarial training and ELLE are instead initialized using PyTorch's default initialization.

All the `PreActResNet18` experiments use a short schedule popular in the literature (Andriushchenko & Flammarion, 2020; Wong et al., 2020; de Jorge et al., 2022; Rocamora et al., 2024). The batch size is set to 128, and SGD with weight decay of 5×10^{-4} is used for the optimization. Crucially, no gradient clipping is employed, which (in addition to network depth and the lack of ramping up) we found to be a major factor behind the instability of pure IBP training (see §3.2). On CIFAR-10 and CIFAR-100 we train a PreActResnet18 for 30 epochs with a cyclic learning rate linearly increasing from 0 to 0.2 during the first half of the training then decreasing back to 0. On SVHN the training is done for 15 epochs, with a cyclic learning rate linearly increasing from 0 to 0.05 during 6 epochs, then decreasing back to 0 for the remaining 9 epochs. Furthermore, for SVHN only, the attack perturbation radius is ramped up from 0 to ϵ during the first 5 epochs. For consistency, the `CNN` experiments from figures 8 and 9 employ the same training schedule.

All the MTL-IBP and ForwAbs experiments using the cyclic schedule gradually increase both the method coefficient (respectively α and λ) and the perturbation radius used to compute the IBP bounds (or their proxy $\bar{\boldsymbol{\delta}}^n$) from 0 to their target value. We conduct an ablation study on MTL-IBP in the context of the experiment from figure 2 to justify this choice: we refer the reader to appendix D.1. The increase happens over the first 25 epochs for CIFAR-10 and CIFAR-100, and over the first 12 epochs for SVHN. The radius used to compute the attack is kept consistent with the adversarial training literature (that is, constant in all cases except for the first 5 epochs on SVHN). In particular, the value is increased exponentially for the first 25% of the above epochs and linearly for the rest (Shi et al., 2021), relying on the `SmoothedScheduler` from `auto_LiRPA` (Xu et al., 2020). In all cases, the attack perturbation radius is left unchanged with respect to the schedules described above for consistency with pure adversarial training.

The long training schedule used for the experiments in table 1 mirror a setup from Shi et al. (2021), widely adapted in the certified training literature (Müller et al., 2023; Mao et al., 2023; De Palma et al., 2024b). Training is carried out for 160 epochs using the Adam optimizer is employed with a learning rate of 5×10^{-4}, decayed twice by a factor of 0.2 at epochs 120 and 140. Gradient clipping is employed, with the maximal ℓ_2 norm of gradients equal to 10. During the first 80 epochs, the perturbation radius (regardless of the computation it is employed for) is increased from 0 to its target value using the `SmoothedScheduler` from `auto_LiRPA` (Xu et al., 2020). For Exp-IBP, we employ a specialized regularizer (Shi et al., 2021) for the IBP bounds for the first 80 epochs (with coefficient 0.5 as done by the original authors on CIFAR-10), as commonly done in previous work (Müller et al., 2023; Mao et al., 2023; De Palma et al., 2024b).

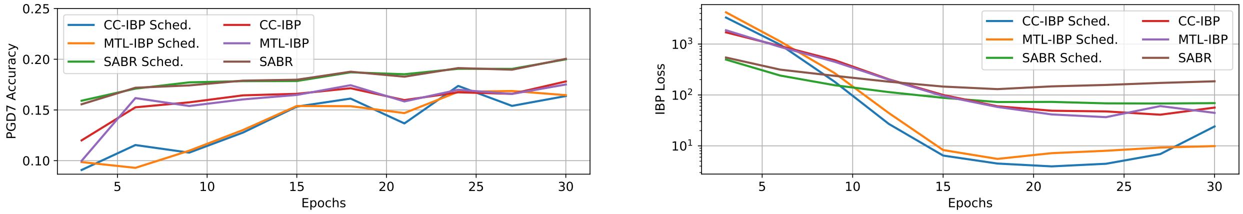
C.3.3 Hyper-parameters

Experiment	Exp-IBP α	MTL-IBP α	ForwAbs $\tilde{\lambda}$
Figures 6a, 18a and 19a	2×10^{-3}	1×10^{-8}	1×10^{-26}
Figures 6b, 18b and 19b	2×10^{-2}	1×10^{-8}	1×10^{-24}
Figures 7a, 18h and 19c	5×10^{-3}	1×10^{-8}	1×10^{-18}
Figures 7b, 18d and 19d	2×10^{-3}	2.75×10^{-15}	1×10^{-23}
Figures 8b, 18f and 19f	4×10^{-1}		1×10^{-8}
Figures 9b, 18f and 19h	2×10^{-3}		1×10^{-10}
Figures 8a, 18e and 19e	2×10^{-1}		1×10^{-10}
Figures 9a, 18g and 19g	2×10^{-3}		1×10^{-12}
Table 1, CIFAR-10, $\epsilon = {}^{8}/_{255}$	5×10^{-3}		1×10^{-12}
Table 1, CIFAR-10, $\epsilon = {}^{24}/_{255}$ and $\epsilon = {}^{16}/_{255}$	7.5×10^{-1}		1×10^{-8}
Table 1, CIFAR-100, $\epsilon = {}^{24}/_{255}$	7.5×10^{-1}		1×10^{-8}

Table 2: Exp-IBP, MTL-IBP and ForwAbs coefficients for figures 6 to 9, figure 18, figure 19, and table 1.

For N-FGSM, we use the same hyper-parameters as de Jorge et al. (2022): the uniform perturbation is sampled from $[-2\epsilon, 2\epsilon]$ for every setting except for SVHN with $\epsilon = 12/255$ where the perturbation is sampled from $[-3\epsilon, 3\epsilon]$. The step size is set to $\alpha = \epsilon$ for all settings. For PGD baselines, we use a step size of $\alpha = {}^{\epsilon}/_{4}$ in all settings. Unless stated otherwise, all the certified training algorithms rely on N-FGSM to generate the adversarial attack composing their loss. To mirror the hyper-parameter in our codebase, we will report $\tilde{\lambda} = {}^{\lambda}/_{2}$ for the employed ForwAbs regularization coefficient λ.

All α and λ parameters for Exp-IBP, MTL-IBP and ForwAbs Figures 6 to 9, 18 and 19 and table 1 are reported in table 2. Exp-IBP PGD5 on CIFAR-10 with ${}^{24}/_{255}$ from table 1 uses $\alpha = 0.75$. All the tuning for these experiments is done on a random validation set (see appendix B.2). On all experiments except those in figure 7, tuning is carried out by maximizing empirical robustness while above the CO threshold.



(a) Empirical accuracy under PGD7 attacks.

(b) IBP loss.

Figure 13: Effect of scheduling the bounding perturbation radius and α on the IBP certified robustness attained by CC-IBP, MTL-IBP and SABR on the `PreActResNet18` training setup from de Jorge et al. (2022). Validation results on CIFAR-10 under perturbations of $\epsilon = 8/255$.

Except where otherwise stated, the employed metric is validation AA accuracy. As explained in §5.1, certified training schemes in figure 7 were tuned using 5 seeds, maximizing PGD-50 robustness while excluding from the final hyper-parameter values any configuration displaying a strong CO behavior.

For the experiments of §3.2, §3.3 and appendix D.1, whose goal was instead to reduce the IBP loss as much as possible on the validation set, we employed the following α values: $\alpha = 0.1$ for Exp-IBP, $\alpha = 10^{-6}$ for CC-IBP and MTL-IBP with scheduling, $\alpha = 10^{-15}$ for CC-IBP and MTL-IBP without scheduling, $\alpha = 10^{-4}$ for SABR with scheduling and $\alpha = 10^{-9}$ for SABR without scheduling. Similarly, in figure 4, $\tilde{\lambda} = 10^{-15}$ was used for ForwAbs and $\lambda_{\ell_1} = 0.04$ for ℓ_1-regularized N-FGSM. For all methods in this experiment, larger values among those we considered led to numerical problems or trivial behaviors, such as networks consistently outputting the same class in our implementation).

Finally, figure 5 employs the following hyper-parameters: on $\epsilon = 8/255$, $\alpha = 3 \times 10^{-2}$ for Exp-IBP, $\alpha = 10^{-8}$ for MTL-IBP, $\tilde{\lambda} = 10^{-18}$ for ForwAbs; on $\epsilon = 24/255$, $\alpha = 2 \times 10^{-2}$ for Exp-IBP, $\alpha = 10^{-7}$ for MTL-IBP, $\tilde{\lambda} = 2 \times 10^{-16}$ for ForwAbs. The goal of the experiment is to qualitatively discern whether certified training schemes can prevent catastrophic overfitting on FGSM. In order to determine this, we ran a series of Exp-IBP, MTL-IBP and ForwAbs experiments directly on the CIFAR-10 test set, and plotted the successful runs (in terms of preventing CO) with the smallest α or $\tilde{\lambda}$ values (among those tried).

In conclusion of this appendix, we show a preliminary version of figure 8b resulting from insufficient tuning of the ForwAbs coefficient λ. Specifically, when tuning for the experiment, we first focused on excessively large coefficients, selecting $\tilde{\lambda} = 10$ as the best-performing setup in terms of validation AA accuracy (as opposed to $\tilde{\lambda} = 10^{-8}$ used for figure 8b). Figure 12 shows the result: while ForwAbs achieves large IBP accuracy on `CNN-5`, further confirming its utility towards certified robustness, it attains low empirical robustness throughout the experiment. Noticing instead the strong empirical robustness of ForwAbs with $\tilde{\lambda} = 10^{-10}$ on `CNN-7` (figure 8a), we extended the tuning on `CNN-5` to smaller $\tilde{\lambda}$ values. We then found $\tilde{\lambda} = 10^{-8}$ to yield stronger validation AA accuracy than $\tilde{\lambda} = 10$, hence producing figure 8b.

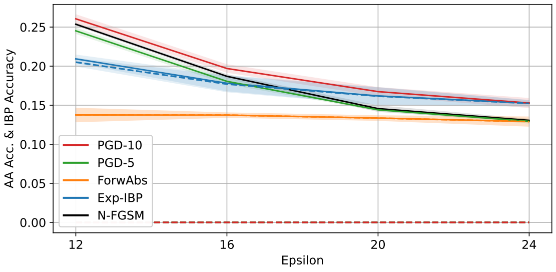


Figure 12: Results of insufficient tuning of ForwAbs for `CNN-5` on CIFAR-10, setup of figure 8b.

D Additional Experimental Results

This appendix displays supplementary experimental results.

D.1 Supplement to Section 3.2, Scheduling Ablation

We here present experimental results omitted from §3.2 and appendix A. Figure 14 displays the behavior of the training IBP loss for Exp-IBP. In particular, it shows that the training loss drops from more than 10^{17} to 2.67 from the first to the second epoch, demonstrating the large degree of variability in IBP loss for the employed `PreActResNet18` training setup. In this context, as shown in figure 3, Exp-IBP displays less sensitivity to the magnitude of the IBP loss, proving more effective for the purposes of decreasing the IBP loss. In fact, the other losses from equation (5) display numerical problems that prevent successful training for large α coefficients (and for pure IBP, which corresponds to $\alpha = 1$ for any expressive loss), leading to aborted runs in our `PyTorch` implementation.

Figure 13 studies the effect of scheduling (gradually increasing from 0 to their target value in 25 epochs) both α and the perturbation radius ϵ employed to compute the IBP lower bounds $\boldsymbol{l}_{\boldsymbol{f_\theta}}^{\mathcal{B}_\epsilon(\boldsymbol{x}),y}$. Excluding trivial outcomes, such as networks systematically outputting the same class, for CC-IBP, MTL-IBP and SABR, we were unable to reach validation IBP loss values below 44 without scheduling. The scheduling allows the use of larger α values, resulting in lower IBP loss value for CC-IBP, MTL-IBP and SABR: respectively from 56.47 to 24.06, from 44.50 to 9.89, and from 185.51 to 69.09. These improvements come with a slight decrease in robustness to PGD-7 (a PGD attack with 7 steps). Nevertheless, CC-IBP, MTL-IBP and SABR are unable to attain non-negligible certified accuracy via IBP in this context. In view of these results, we employ the same scheduling for ForwAbs (see appendix C.3.2).

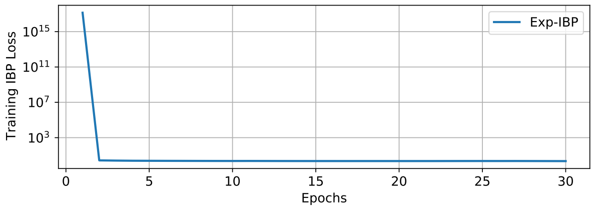


Figure 14: Training IBP loss of the Exp-IBP experiment from figure 2.

D.2 Training Overhead

Figure 15 shows the training overhead of MTL-IBP, Exp-IBP, ForwAbs, ELLE-A, PGD-5 and PGD-10 with respect to N-FGSM (which is used to compute the attack for MTL-IBP, Exp-IBP, ForwAbs and ELLE-A) when training a `PreActResNet18` on 80% of the CIFAR-10 training set, complementing the information provided in figure 4. As described in §3.3, ForwAbs has minimal overhead with respect to N-FGSM. On the other hand, MTL-IBP and Exp-IBP display a runtime slightly smaller than PGD-5, with ELLE-A being only marginally less expensive than both of them. While the original work (Rocamora et al., 2024) reports less overhead for ELLE-A, which requires 3 batched forward passes to compute its regularization term, we found its overhead to be heavily dependent on the GPU model, with newer models leading to a faster execution of the batched forward pass (the original work relies on an Nvidia A100 SXM4 GPU, whereas our timing experiments are carried out on an older Nvidia GTX 1080Ti GPU). As expected, PGD-10 training requires almost twice the time as PGD-5, and more than half the time as MTL-IBP and Exp-IBP.

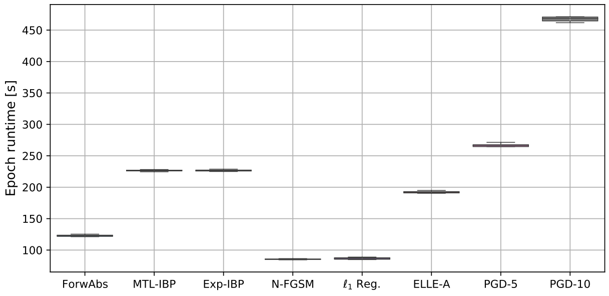
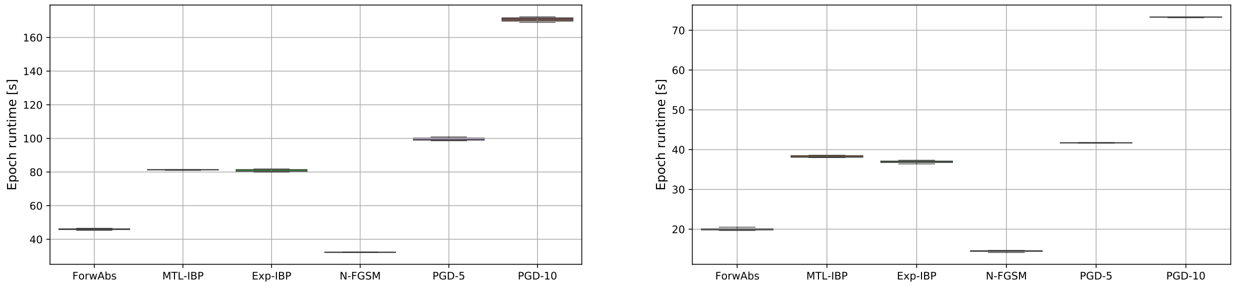


Figure 15: Box plots (10 repetitions) for the CIFAR-10 training time of a single epoch (using a 80% subset of the training set) on the `PreActResNet18` setup from de Jorge et al. (2022).

Figure 16, instead, plots the training overheads when training `CNN-7` and `CNN-5`. The relative runtimes across methods remain similar to `PreActResNet18`, with the runtime of each algorithm markedly smaller on the `CNN` models, and for the 5-layer network.



(a) `CNN-7`.

(b) `CNN-5`.

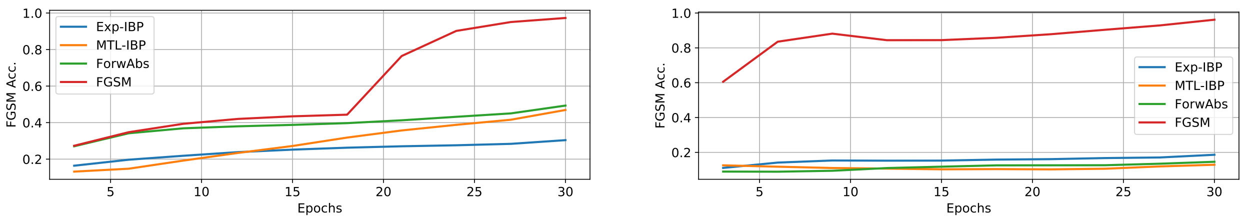
Figure 16: Box plots (10 repetitions) for the CIFAR-10 training time of a single epoch (using a 80% subset of the training set) on `CNN-7` and `CNN-5`.

D.3 Supplement to FGSM experiments

Figure 17 plots the training adversarial accuracy computed via FGSM, the attack employed to generate $\boldsymbol{x}_{\text{adv},\mathbb{A}}$ for all training methods in this experiment, associated to the plots in figure 5. The behavior of the training adversarial accuracy for FGSM, which spikes up for $\epsilon = {}^{8}/_{255}$ roughly when its validation PGD-7 accuracy goes towards 0 (cf. figure 5), and attains very large values for $\epsilon = {}^{24}/_{255}$ in spite of its null validation PGD-7 accuracy, is a clear marker of catastrophic overfitting.

D.4 Clean Accuracies and IBP Loss for N-FGSM Experiments

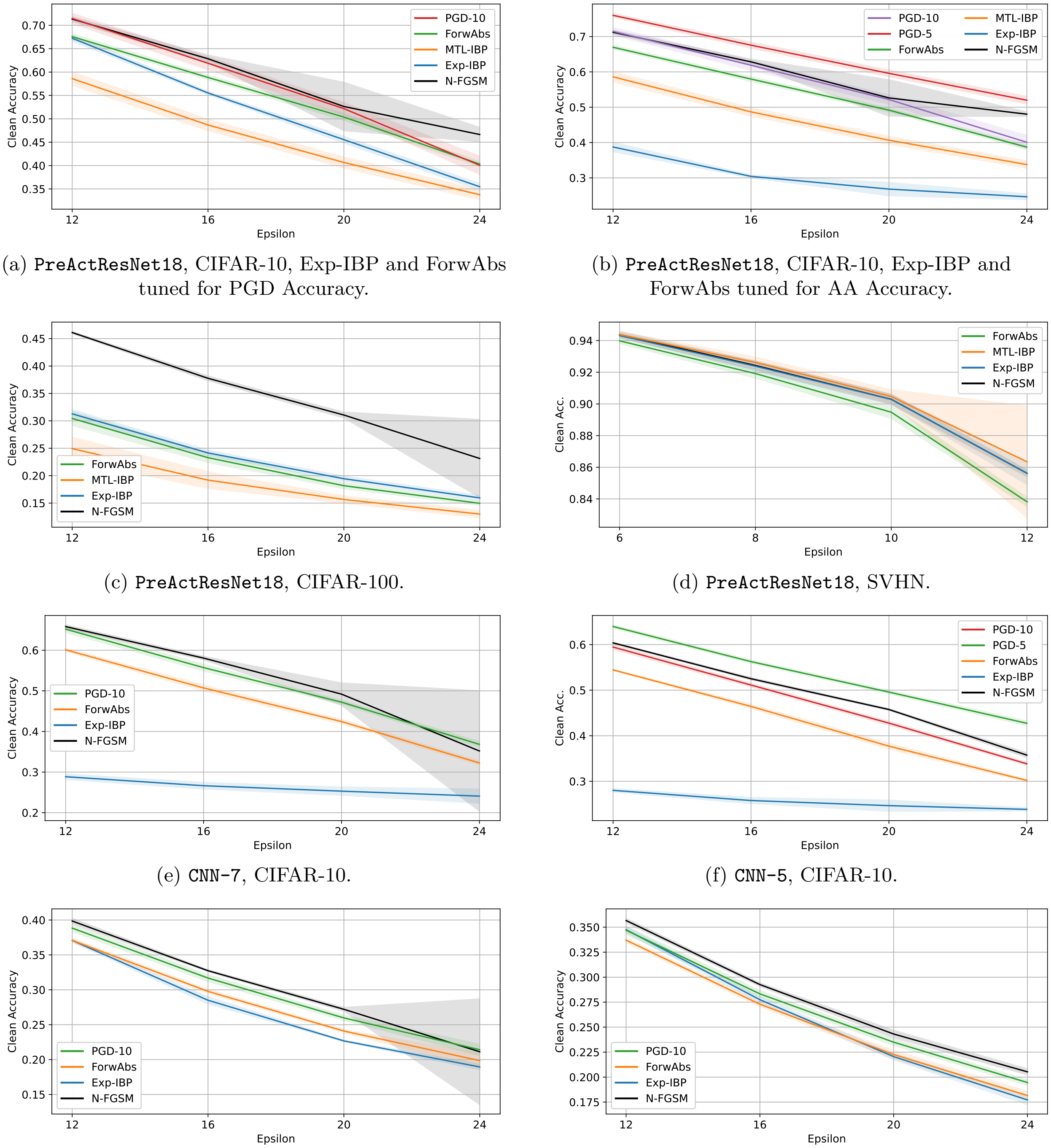
We here provide omitted clean accuracies and IBP losses for the experiments for figures 6 to 9 and table 1. Figure 18 provides the results for the cyclic training schedule on `PreActResNet18`, `CNN-7` and `CNN-5`. It is clear that the positive effects associated to the certified training schemes (the mitigation or prevention of catastrophic overfitting in §5.1 and the empirical robustness improvements detailed in §5.2) come at a cost in clean accuracy, which is a limitation of these methods and of enforcing low IBP loss. Generally speaking, as visible in figure 19, Exp-IBP attains significantly better trade-offs between clean accuracy and IBP loss, especially when compared to MTL-IBP. As a result, Exp-IBP is to be preferred if IBP accuracy is the primary metric. Table 3 presents the clean accuracies and IBP losses pertaining to the long schedule experiment from table 1, additionally presenting a comparison between the IBP losses of the certified training schemes tuned for empirical robustness and the adversarial training baselines whose IBP loss was shown in figure 1. In all cases, certified training schemes decrease the IBP loss compared to the adversarial training baselines at a cost in standard accuracy, with a larger cost associated to a greater reduction in IBP loss. We conclude by pointing out that an alternative tuning criterion (considering trade-offs between standard accuracy and empirical robustness) could result in more favorable trade-offs. This was not the focus of the presented experiments.



(a) FGSM training accuracy for $\epsilon = {}^{8}/_{255}$.

(b) FGSM training accuracy for $\epsilon = {}^{24}/_{255}$.

Figure 17: Training FGSM accuracy for the experiments from figure 5.



(a) `PreActResNet18`, CIFAR-10, Exp-IBP and ForwAbs tuned for PGD Accuracy.

(b) `PreActResNet18`, CIFAR-10, Exp-IBP and ForwAbs tuned for AA Accuracy.

(c) `PreActResNet18`, CIFAR-100.

(d) `PreActResNet18`, SVHN.

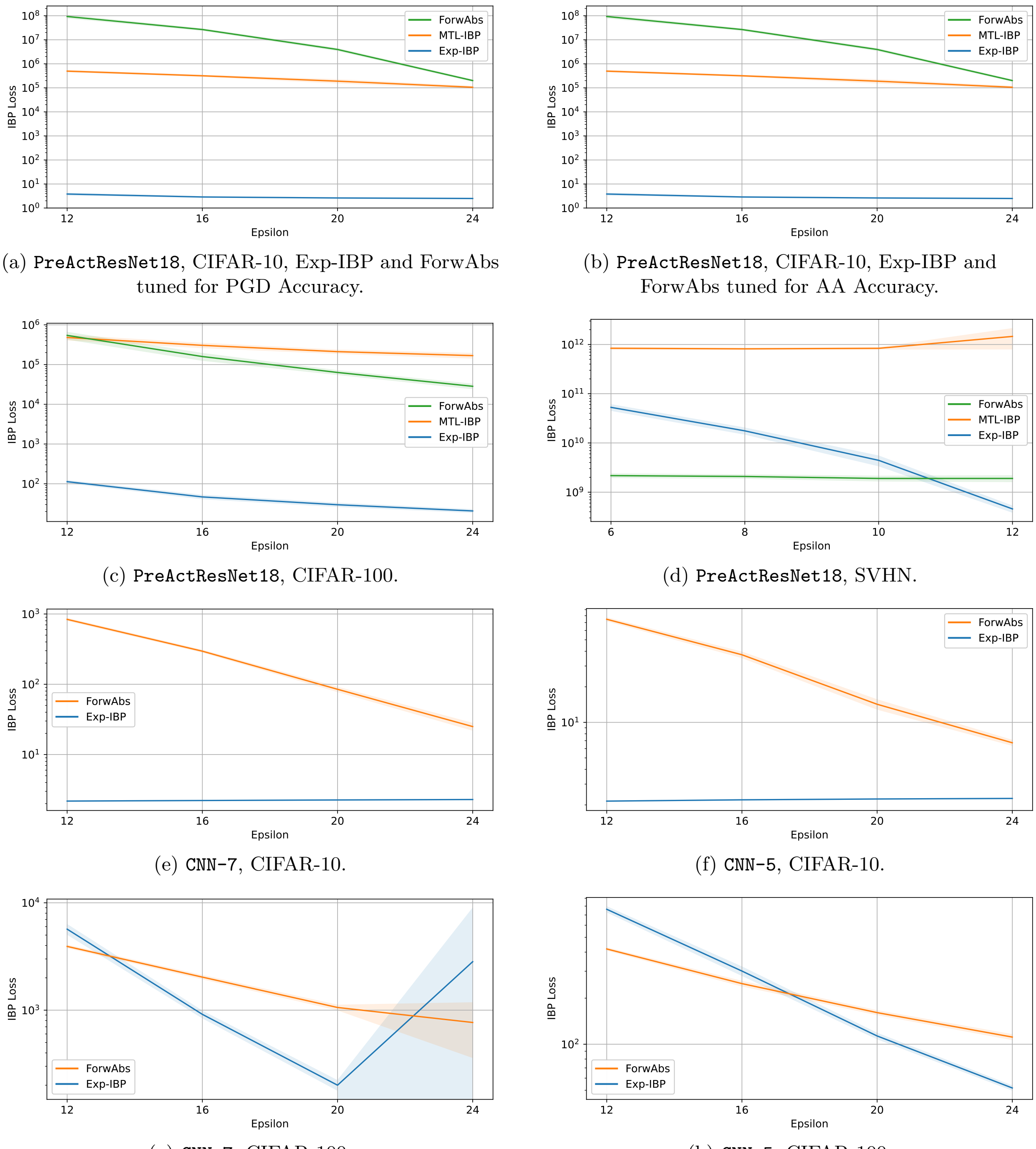
(e) `CNN-7`, CIFAR-10.

(f) `CNN-5`, CIFAR-10.

(g) `CNN-7`, CIFAR-100.

(h) `CNN-5`, CIFAR-100.

Figure 18: Clean accuracies for the experiments from figures 6, 7, 8 and 9. Means and 95% confidence intervals over 5 repetitions.



(a) PreActResNet18, CIFAR-10, Exp-IBP and ForwAbs tuned for PGD Accuracy.

(b) PreActResNet18, CIFAR-10, Exp-IBP and ForwAbs tuned for AA Accuracy.

(c) PreActResNet18, CIFAR-100.

(d) PreActResNet18, SVHN.

(e) CNN-7, CIFAR-10.

(f) CNN-5, CIFAR-10.

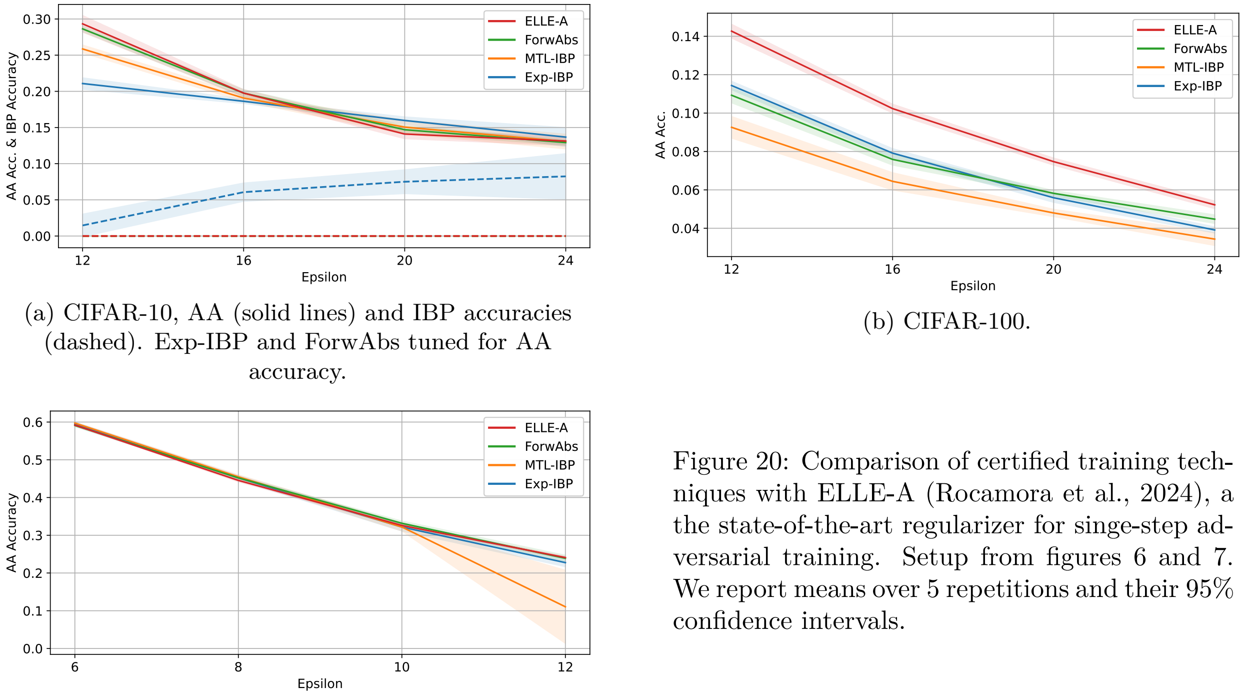
(g) CNN-7, CIFAR-100.

(h) CNN-5, CIFAR-100.

Figure 19: IBP losses of methods from §3 for the experiments from figures 6, 7, 8 and 9. Means and standard deviations over 5 repetitions.

Table 3: Clean accuracies and IBP losses for the experiments in table 1.

Dataset	ϵ	Method	Clean acc. [%]	IBP loss
CIFAR-10	$\frac{8}{255}$	Exp-IBP	76.30 ± 0.43	$(4.90 \pm 0.55) \times 10^4$
		IBP	39.37 ± 1.51	1.92 ± 0.01
		ForwAbs	74.26 ± 0.28	$(2.05 \pm 0.11) \times 10^4$
		PGD-10	77.07 ± 0.31	$(9.79 \pm 0.85) \times 10^5$
		PGD-5	79.81 ± 0.38	$(1.72 \pm 0.11) \times 10^6$
		N-FGSM	77.28 ± 0.26	$(1.91 \pm 0.14) \times 10^6$
	$\frac{16}{255}$	Exp-IBP	31.30 ± 2.13	2.15 ± 0.02
		IBP	31.65 ± 0.55	2.13
		ForwAbs	46.54 ± 0.31	$(7.72 \pm 0.37) \times 10^1$
		PGD-10	58.44 ± 0.24	$(3.57 \pm 0.33) \times 10^5$
		PGD-5	64.07 ± 0.14	$(1.01 \pm 0.13) \times 10^6$
		N-FGSM	62.38 ± 9.16	$(2.93 \pm 2.66) \times 10^6$
	$\frac{24}{255}$	Exp-IBP	23.03 ± 4.97	2.25 ± 0.02
		Exp-IBP PGD-5	27.32 ± 2.94	2.23 ± 0.02
		IBP	23.51 ± 2.63	2.22 ± 0.01
		ForwAbs	30.15 ± 0.37	8.73 ± 0.53
		PGD-10	39.99 ± 0.60	$(2.60 \pm 0.42) \times 10^4$
		PGD-5	49.31 ± 0.37	$(2.13 \pm 0.22) \times 10^5$
		N-FGSM	57.46 ± 9.60	$(3.31 \pm 1.55) \times 10^6$
CIFAR-100	$\frac{24}{255}$	Exp-IBP	6.85 ± 1.29	4.49 ± 0.02
		IBP	7.29 ± 0.61	4.45 ± 0.01
		ForwAbs	16.85 ± 0.67	$(5.79 \pm 0.36) \times 10^1$
		PGD-10	23.56 ± 0.61	$(2.19 \pm 0.19) \times 10^5$
		PGD-5	28.53 ± 0.46	$(6.12 \pm 0.32) \times 10^5$
		N-FGSM	29.07 ± 3.01	$(6.50 \pm 1.92) \times 10^6$



(a) CIFAR-10, AA (solid lines) and IBP accuracies (dashed). Exp-IBP and ForwAbs tuned for AA accuracy.

(b) CIFAR-100.

(c) SVHN.

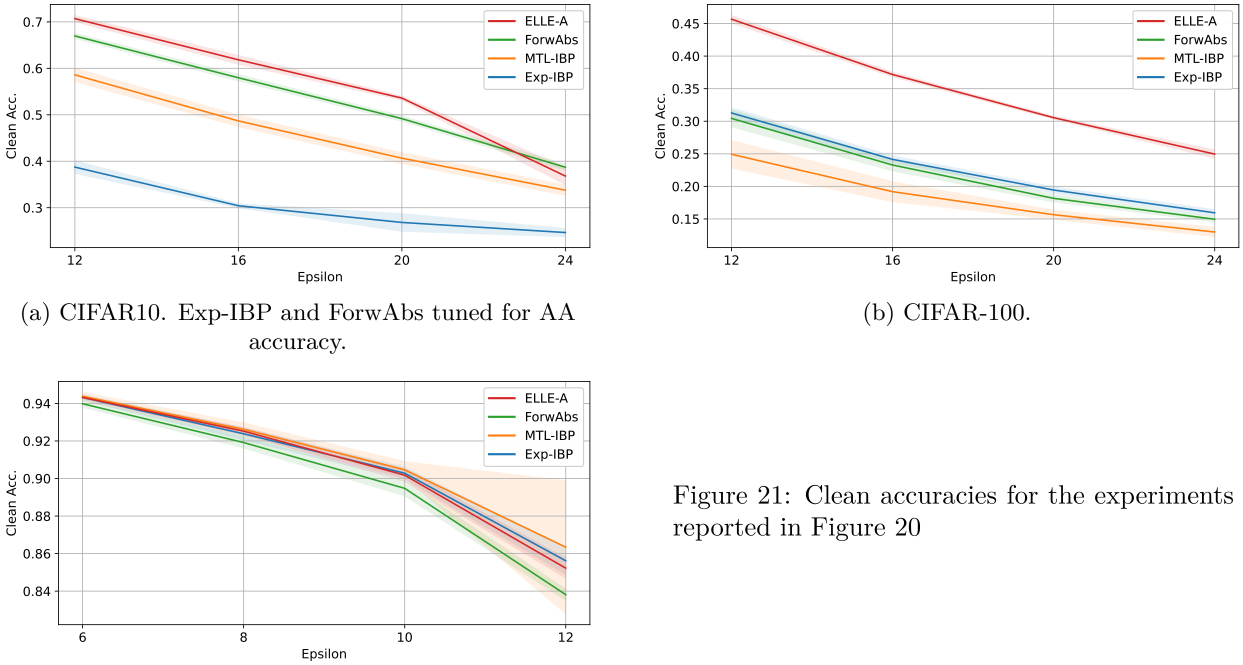
Figure 20: Comparison of certified training techniques with ELLE-A (Rocamora et al., 2024), a the state-of-the-art regularizer for singe-step adversarial training. Setup from figures 6 and 7. We report means over 5 repetitions and their 95% confidence intervals.

D.5 Comparison with ELLE

The main objective of this work is to demonstrate that certified training techniques can be successfully employed towards empirical robustness, hence beyond their original design goal. As part of the provided evidence, section 5.1 shows that Exp-IBP and ForwAbs can prevent CO on settings common to the single-step adversarial training literature. In order to contextualize their performance for the task, this appendix provides a comparison of the performance of Exp-IBP, MTL-IBP and ForwAbs with ELLE-A (Rocamora et al., 2024), a state-of-the-art method designed to prevent catastrophic overfitting.

The experimental setup is the one from figures 6 and 7 (a `PreActResNet18` trained with a cyclic training schedule). As for the certified training techniques, we use N-FGSM as underlying adversarial attack. The runtime overhead of ELLE-A, which is slightly smaller than MTL-IBP and Exp-IBP and significantly larger than ForwAbs on an Nvidia GTX 1080Ti GPU, is reported in figure 15. We tuned the ELLE-A hyper-parameter λ_{ELLE}, which controls the amount of regularization imposed on a notion of local linearity proposed by the authors, consistently with the way MTL-IBP, Exp-IBP and ForwAbs were tuned for figures 6 and 7. The tuning resulted in the following values: $\lambda_{\text{ELLE}} = 6000$ for CIFAR-10 (which maximizes both the validation PGD-50 and AA accuracy), $\lambda_{\text{ELLE}} = 3000$ for CIFAR-100, and $\lambda_{\text{ELLE}} = 1000$ for SVHN.

As reported in Figure 20, ELLE-A prevents CO across all settings. Remarkably, on CIFAR-10, certified training techniques match or outperform ELLE-A in AA accuracy for $\epsilon \geq {}^{20}/_{255}$, with Exp-IBP outperforming ELLE-A while also producing non-negligible certified robustness via IBP. We believe this suggests that verifiability and empirical robustness are not conflicting objectives in this setup. Furthermore, ForwAbs performs at least competitively with ELLE-A on all the considered epsilons while reducing its overhead. The situation is drastically different on the harder CIFAR-100, where ELLE-A markedly outperforms certified training schemes. We speculate the employed network may lack the capacity to sustain tight over-approximations while preserving empirical robustness. On SVHN, except MTL-IBP which fails owing to the sensitivity of its loss as discussed in §5.1, all the algorithms attain similar performance profiles, with the most robust algorithm depending on the perturbation radius. Figure 21 reports the corresponding clean accuracies: on average ELLE-A has less impact on clean accuracy than certified training schemes, except for SVHN, where expressive losses display larger standard performance, and on CIFAR-10 for $\epsilon = {}^{24}/_{255}$, where ForwAbs does.



(a) CIFAR10. Exp-IBP and ForwAbs tuned for AA accuracy.

(b) CIFAR-100.

(c) SVHN.

Figure 21: Clean accuracies for the experiments reported in Figure 20

In summary of this appendix, we argue that, on easier datasets such as CIFAR-10, certified training techniques such as Exp-IBP and ForwAbs should be considered as strong baselines for CO prevention by the single-step adversarial training community.

E Previous Work on the Empirical Accuracy of Certified Training

We now comment on the relationship between §5.2 with literature results on the empirical accuracy of certified training algorithms (see §4).

While this was not the focus of their works, Gowal et al. (2018); Mao et al. (2024a) both demonstrated that certified training schemes (including IBP for Gowal et al. (2018), IBP, SABR and MTL-IBP for Mao et al. (2024a)) outperform multi-step adversarial training on large perturbation radii on MNIST for relatively shallow convolutional networks. Specifically, Gowal et al. (2018) showed that IBP outperforms PGD-7 training in terms of PGD-200-10 (PGD with 200 iterations and 10 restarts) accuracy for $\epsilon \in \{0.3, 0.4\}$, when using a 100-epoch training schedule. Mao et al. (2024a) tuned the certified training schemes to maximize certified robustness, and then positively compared their empirical robustness, measured through the attacks within MN-BaB, a complete verifier based on branch-and-bound (Ferrari et al., 2022), to the AA accuracy of PGD-5-3 training for $\epsilon = 0.3$ (using a 70-epoch training schedule). We omit MNIST from our study owing to its relative simplicity, and to our focus on settings typically considered in the single-step adversarial training literature (de Jorge et al., 2022; Rocamora et al., 2024; Andriushchenko & Flammarion, 2020).

Furthermore, both works reported that certified training schemes (IBP for Gowal et al. (2018), SABR and MTL-IBP for Mao et al. (2024a)) can outperform multi-step attacks in terms of empirical robustness using `CNN-7` for $\epsilon = {}^{8}/_{255}$ on CIFAR-10. Gowal et al. (2018) employ a 3200-epoch training schedule, using which PGD-7 and IBP respectively attain 34.77% and 24.95% PGD-200-10 accuracy. Mao et al. (2024a) rely instead on a 240-epoch schedule and again tune certified training schemes to maximize certified robustness, reporting 35.93% AA accuracy for PGD-10-3, 36.11% and 36.02% MN-BaB empirical robustness for SABR and MTL-IBP, respectively. We remark that N-FGSM already attains 37.76% average AA accuracy using the same network architecture in table 1, outperforming the results reported above for both adversarial and certified training, and highlighting the importance of strong baselines and appropriate regularization. In fact, differently from Mao et al. (2024a), who adopt ℓ_1 regularization also for the adversarial training baselines, we use weight decay (with coefficient 5×10^{-5}), which is more commonly adopted in the adversarial training literature to combat robust overfitting.

Differently from the previous literature, this work is characterized by a strong focus on preventing catastrophic overfitting (§5.1), which is novel to the best of our knowledge. Section 5.2, which instead examines settings where certified training schemes relying on single-step attacks can overcome multi-step attacks, complements and extends the aforementioned literature results. Crucially, rather than simply assessing the robustness of certified training schemes deployed to maximize certified robustness, we also tune these algorithms *for* empirical robustness. This can make a difference in practice. For instance, Exp-IBP attains 38.44% average AA accuracy in our experiments for $\epsilon = {}^{8}\!/_{255}$ on CIFAR-10, outperforming N-FGSM and PGD-5 from table 1, and all values reported by Mao et al. (2024a). Section 5.2 provides a fair and systematic comparison (relying on the same strong attack suite (Croce & Hein, 2020), which is also used as tuning metric on validation) which includes larger perturbation radii (up to $\epsilon = {}^{24}\!/_{255}$), the harder CIFAR-100 dataset, and a shorter cylic training schedule more common in the adversarial training literature.